



Universidad
Tecnológica
de Pereira

MAESTRÍA EN INGENIERÍA DE SISTEMAS
LÍNEA CIENCIAS COMPUTACIONALES

CONSTRUCCIÓN DE UN OPERADOR DE MATCHING ONTOLÓ-
GICO PARA LOS NIVELES DE REPRESENTACIÓN SEMÁN-
TICO Y SEMIÓTICO

TESIS DE MAESTRÍA PROPUESTA POR

ING. ANDRÉS VARGAS GARCÍA

DIRIGIDA POR:

ING. JULIO CESAR CHAVARRO PORRAS PhD

PEREIRA 2017



Universidad
Tecnológica
de Pereira

MAESTRÍA EN INGENIERÍA DE SISTEMAS
LÍNEA CIENCIAS COMPUTACIONALES

CONSTRUCCIÓN DE UN OPERADOR DE MATCHING ONTOLÓ-
GICO PARA LOS NIVELES DE REPRESENTACIÓN SEMÁN-
TICO Y SEMIÓTICO

TESIS QUE PRESENTA:

ING. ANDRÉS VARGAS GARCÍA

Para obtener el grado de:

MAESTRO EN CIENCIAS

Director De La Tesis:

ING. JULIO CESAR CHAVARRO PORRAS PhD

PEREIRA, COLOMBIA

DICIEMBRE DE 2017

© Copyright by
Andrés Vargas García
2017

La tesis de Andrés Vargas García es aprobada por:

Pereira, Risaralda, Colombia., _____ de _____.

A mi familia por su apoyo incondicional.

i. CONTENIDO

i.	CONTENIDO.....	I
ii.	LISTA DE FIGURAS.....	VI
iii.	LISTA DE TABLAS	VIII
iv.	ACRÓNIMOS, SIGLAS Y DEFINICIONES	IX
v.	AGRADECIMIENTOS	XVI
vi.	RESUMEN	XVII
1	INTRODUCCIÓN	1
1.1	PRESENTACIÓN	1
1.2	DEFINICIÓN DEL PROBLEMA.....	5
1.3	MOTIVACIÓN	7
1.4	ALCANCES Y CONTRIBUCIONES.....	8
1.5	HIPÓTESIS Y METODOLOGÍA DE INVESTIGACIÓN	9
1.6	ESTRUCTURA DE LA TESIS	11
2	OBJETIVOS	13
2.1.	OBJETIVOS ESPECÍFICOS	13
3	FUNDAMENTOS TEÓRICOS, TECNOLOGÍAS Y ESTADO DEL ARTE	15
I		

3.1	ONTOLOGÍA	15
3.1.1.	CONCEPTO DE ONTOLOGIA	16
3.1.2.	DEFINICION DE ONTOLOGI.....	19
3.1.3.	CLASIFICACION DE ONTOLOGIAS	22
3.2	CAMBIO ONTOLÓGICO	27
3.2.1.	RESUMEN DE CAMBIOS ONTOLOGICOS.....	33
3.3	REPRESENTACIÓN DE UNA ALINEACIÓN	38
3.4	<i>MATCHING</i> ONTOLÓGICO	40
3.4.1.	EL PROCESO DE <i>MATCHING</i>	41
3.4.2.	CLASIFICACION DE LAS TECNICAS DE <i>MATCHING</i>	43
3.4.3.	ESTRATEGIAS DE <i>MATCHING</i>	49
3.4.4.	ESTRATEGIAS DE COMBINACION DE TECNICAS	51
3.5	EL MODELO CONCEPTUAL	54
3.5.1.	COMPONENTE ESTRUCTURAL DEL MODELO CONCEPTUAL	56
3.5.2.	REPRESENTACIÓN ESTRUCTURAL DE LA ONTOLOGÍA MEDIANTE UN GRAFÁRBOL	57
3.5.3.	LOS OPERADORES DE CAMBIO ONTOLÓGICO	63
3.5.4.	ESPECIFICACIÓN ESTRUCTURAL DEL LENGUAJE DECLARATIVO	66

4.	DISEÑO DEL OPERADOR DE <i>MATCHING</i>	78
4.1.	DISEÑO DE LA ARQUITECTURA DEL MATCHER	80
4.2.	ESPECIFICACIÓN ESTRUCTURAL DEL MATCHER	82
4.2.1.	OPERACIONES DE PREPROCESAMIENTO (OPP).....	84
4.2.2.	OPERACIONES DE PROCESAMIENTO DE <i>MATCHING</i> 86	
4.2.3.	OPERACIONES DE COMBINACION.....	90
4.2.4.	OPERACIONES DE FILTRO	93
4.3.	MODELO TECNOLÓGICO	97
5.	CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS...	105
5.1.	CONCLUSIONES	105
5.2.	TRABAJOS FUTUROS	108
6.	ANEXOS.....	110
6.1.	SISTEMAS DE <i>MATCHING</i>	110
6.1.1.	COMA.....	110
6.1.2.	GLUE	111
6.1.3.	S-MATCH	113
6.1.4.	OLA.....	116
6.1.5.	FALCON AO.....	118

6.1.6.	SAMBO.....	120
6.1.7.	AROMA	122
6.1.8.	RIMON	125
6.1.9.	AGREEMENT MAKER	126
6.1.10.	ASMOV.....	128
6.1.11.	TAXOMP.....	130
6.1.12.	PROMPT	131
6.1.13.	DSSIM.....	132
6.2.	DESCRIPCIÓN DE TÉCNICAS DE <i>MATCHING</i> BÁSICAS..	134
6.2.1.	TÉCNICAS SINTÁCTICAS BASADAS EN CADENAS	134
6.2.2.	DISTANCIA ENTRE CADENAS	136
6.2.3.	SIMILITUD ENTRE SUBCADENAS.....	138
6.2.4.	TÉCNICAS SINTÁCTICAS BASADOS EN RESTRICCIONES	139
6.2.5.	COMPARACIÓN DE TIPOS DE DATOS	140
6.2.6.	COMPARACIÓN DE DOMINIO	141
6.2.7.	TECNICAS SINTACTICAS A NIVEL DE ESTRUCTURA 142	
7.	BIBLIOGRAFÍA	144

ii. LISTA DE FIGURAS

FIGURA 1.	ESPECTRO DE ONTOLOGÍAS (GRAFICA SEGÚN (<i>LASSILA & MCGUINNESS, 2001</i>)) ...	23
FIGURA 2.	TIPOS DE ONTOLOGÍA SEGÚN SU NIVEL DE GENERALIDAD (GRAFICA SEGÚN (<i>GUARINO, 1998</i>))	26
FIGURA 3.	EL PROCESO DE <i>MATCHING</i> (BASADO EN (<i>EUZENAT & SHVAIKO, 2007</i>))	43
FIGURA 4.	ESTRUCTURA SECUENCIAL DE <i>MATCHING</i>	52
FIGURA 5.	ESTRUCTURA DE UN MATCHER EN PARALELO CON DOS TÉCNICAS BÁSICAS	53
FIGURA 6.	ESTRUCTURA DE UN MATCHER ITERATIVO.....	54
FIGURA 7.	JERARQUÍA DE RELACIONES (TOMADA DE (<i>CHAVARRO, 2012</i>))	58
FIGURA 8.	GRAFÁRBOL ONTOLÓGICO Y VISTAS POSIBLES (<i>TOMADO DE (CHAVARRO, 2012)</i>)	60
FIGURA 9.	EJEMPLO DE COMPOSICIÓN DINÁMICA	82
FIGURA 10.	ONTOLOGIAS DE CASOS DE PRUEBA, UNIVERSIDAD DE CORNELL Y UNIVERSIDAD DE WASHINGTON.....	83
FIGURA 11.	APLICACIÓN DE LA FUNCION CASENORM	85
FIGURA 12.	EJEMPLO DE MATRIZ DE SIMILITUD	87
FIGURA 13.	EJEMPLO DE MATRIZ DE SIMILITUD SON LOS VALORES SELECCIONADOS PARA LA ALINEACION	87
FIGURA 14.	APLICACIÓN DE LA OPERACIÓN DE PROCESAMIENTO NGRAM	90
FIGURA 15.	EJEMPLO DE CONFIGURACION DE MATCHER	96
FIGURA 16.	ARBOL DE OPERACIONES DEL MATCHER DE EJEMPLO	97
FIGURA 17.	ETAPAS DEL MATCHER	98
FIGURA 18.	ARQUITECTURA MARCO DE REFERENCIA PARA LA GESTIÓN CONCEPTUAL DEL CAMBIO ONTOLÓGICO. TOMADO DE (<i>CHAVARRO, 2012</i>)	99
FIGURA 19.	DIAGRAMA DE COMPONENTES, ADAPTADO DE (<i>SALAZAR, 2015</i>)	100
FIGURA 20.	VISTA INTERNA DEL COMPONENTE GESTOR DE OPERACIONES COMPLEJAS	101
FIGURA 21.	ARQUITECTURA GLUE (TOMADO DE (<i>DOAN ET AL., 2004</i>))	113
FIGURA 22.	ARQUITECTURA DE S-MATCH (TOMADO DE (<i>SHVAIKO ET AL., 2010</i>)).....	114
FIGURA 23.	ARQUITECTURA DEL SISTEMA FALCON-AO (TOMADO DE (<i>JIAN ET AL., 2005</i>))	119

FIGURA 24.	ESTRATEGIA DE ALINEACIÓN DEL FRAMEWORK SAMBO (TOMADO DE (LAMBRIX & TAN, 2011))	122
FIGURA 25.	ARQUITECTURA DEL SISTEMA RIMOM (TOMADO DE (WANG ET AL., 2010))	126
FIGURA 26.	AGREEMENTMAKER OAEI 2010 (TOMADO DE (CRUZ ET AL., 2010))	128
FIGURA 27.	EL PROCESO DE MAPPING DE ASMOV (TOMADO DE (JEAN-MARY ET AL., 2010))	129

iii. LISTA DE TABLAS

TABLA 1. MAPEO ONTOLÓGICO33

TABLA 2. TRANSFORMACIÓN ONTOLÓGICA.....34

TABLA 3. ALINEACIÓN ONTOLÓGICA.....34

TABLA 4. ARTICULACIÓN ONTOLÓGICA35

TABLA 5. TRASLACIÓN ONTOLÓGICA (PRIMER ENFOQUE)35

TABLA 6. TRASLACIÓN ONTOLÓGICA (SEGUNDO ENFOQUE)36

TABLA 7. EVOLUCIÓN ONTOLÓGICA37

TABLA 8. CONTROL DE VERSIONES DE ONTOLOGÍAS37

TABLA 9. INTEGRACIÓN ONTOLÓGICA38

TABLA 10. MIGRACIÓN ONTOLÓGICA.....38

TABLA 11. EJEMPLO DE APLICACIÓN DE OPERACIÓN DE COMBINACIÓN.....93

TABLA 12. EJEMPLO APLICACIÓN OPERACIÓN DE FILTRO95

TABLA 13. TABLA DE COMPATIBILIDAD DE TIPOS 141

iv. ACRÓNIMOS, SIGLAS Y DEFINICIONES

ALIGNMENT

Es el conjunto de correspondencias entre dos o más ontologías. La alineación es la salida del proceso de matching. (*Euzenat & Shvaiko, 2007*)

CERN

El nombre CERN se deriva del acrónimo francés «*Conseil Européen pour la Recherche Nucléaire*», o Consejo Europeo para la Investigación Nuclear, un organismo provisional fundado en 1952 con el mandato de establecer una organización de investigación de física fundamental de clase mundial en Europa. En ese momento, la investigación de la física pura se concentró en la comprensión del interior del átomo, de ahí la palabra "nuclear".(*CERN, 2017*)

En este laboratorio trabajaban los investigadores que dieron origen a lo que hoy conocemos con la Word Wide Web.

INTEGRATION

Es la inclusión en una ontología de otra ontología, expresando las afirmaciones con transparencia entre estas ontologías. La ontología integrada retiene el conocimiento de las dos ontologías integradas.

Contrario a la mezcla, la primera ontología permanece inalterada, mientras que la segunda es modificada (*Euzenat & Shvaiko, 2007*)

MAPPING

Versión dirigida de una alineación, mapea las entidades de una ontología a al menos una entidad de otra ontología. La definición matemática requiere en principio que el objeto maquillado sea igual a su imagen. (*Euzenat & Shvaiko, 2007*)

MATCHER

Es la combinación de técnicas o algoritmos que buscan encontrar las correspondencias entre entidades de dos ontologías, para producir un mecanismo más eficiente y de mejor calidad que las partes que lo conforman. (*Euzenat & Shvaiko, 2007*)

MATCHING

Es el proceso de encontrar relaciones o correspondencias entre entidades de diferentes ontologías. (*Euzenat & Shvaiko, 2007*)

MERGE

Creación de una nueva ontología desde dos ontologías fuente, posiblemente no disjuntas. Las ontologías fuentes permanecen inalteradas,

mientras que la nueva ontología contiene el conocimiento de las ontologías iniciales. (Euzenat & Shvaiko, 2007)

METADATO

Según Howe (Howe, 1993), el término fue acuñado por Jack Myers en la década de los 60's para describir conjuntos de datos. La primera acepción que se le dio (y actualmente la más extendida) fue la de dato sobre el dato, ya que proporcionaban la información mínima necesaria para identificar un recurso. En este mismo trabajo se afirma que puede incluir información descriptiva sobre el contexto, calidad y condición o características del dato. (J. A. Senso & Piñero, 2003)

METAMODELO

Meta modelado o meta-modelos en ingeniería de software e ingeniería de sistemas, entre otras disciplinas, es el análisis, la construcción y el desarrollo de los marcos, reglas, restricciones, modelos y teorías aplicables y útiles para el modelado de una clase predefinida de problemas. (Wikipedia, 2009) Como su nombre lo indica, este concepto se aplica los conceptos de meta-y el modelado. «Meta modelado» es la construcción de una colección de «conceptos» dentro de un determinado dominio. Un modelo es una abstracción de los fenómenos en el mundo real, un metamodelo es otra abstracción, las propiedades del propio modelo de resaltado. Un modelo se ajusta a su metamodelo

en la forma en que un programa de ordenador se ajusta a la gramática del lenguaje de programación en el que está escrito.

ODM

Metamodelo de Definición de Ontologías (Ontology Definition Metamodel), ODM, una especificación que recoge una familia de metamodelos y correspondencias entre ellos, que permiten interoperar con modelos especificados en UML y modelos en otros lenguajes formales como OWL o Common Logic, entre otros. (OMG, 2007)

El ODM sienta las bases para, por ejemplo, poder obtener especificaciones en OWL a partir de modelos UML, y a la inversa, modelos UML a partir de ontologías en OWL. De esta forma, se habilita el desarrollo de ontologías tomando como punto de partida modelos de sistema especificados en UML. En última instancia, la especificación del ODM persigue habilitar el modelado de ontologías mediante herramientas basadas en UML.

OWL

El Lenguaje de Ontologías Web (OWL) diseñado para procesar el contenido de la información en lugar de únicamente representar información para los humanos. OWL facilita un mejor mecanismo de interoperabilidad de contenido Web que los mecanismos admitidos por XML, RDF, y esquema RDF (RDF-S) proporcionando vocabulario adicional junto con una semántica formal. OWL tiene tres sublenguajes,

con un nivel de expresividad creciente: OWL Lite, OWL DL, y OWL Full. (Smith, Welty, & McGuinness, 2004)

RESOURCE DESCRIPTION FRAMEWORK.

RDF (Lassila & Swick, 1998) es una fundación para el procesamiento de metadatos, provee interoperabilidad entre aplicaciones que puede intercambiar información en la web entendible por máquinas. RDF es un modelo de representación de metadatos con una sintaxis para codificar y transportar estos metadatos de una manera que permita maximizar la interoperabilidad entre servidores y clientes web. Uno de los objetivos de RDF es hacer posible la especificación semántica de los datos, basándose en XML como una forma de estandarizar la interoperabilidad, RDF y XML son complementarios.

El objetivo principal de RDF es definir un mecanismo para describir recursos evitando hacer supuestos de un dominio en particular ni define la semántica de un dominio aplicación.

El modelo de datos básicos consiste en tres tipos de objetos: recursos, propiedades, sentencias. Las propiedades pueden ser vistas como atributos de los recursos y en este sentido corresponden a la tradicional dupla atributo-valor, las propiedades también representan relaciones entre recursos.

Un recurso es cualquier cosa que puede ser descrita, por ejemplo, un recurso puede ser una página web completa, una parte de una página web o un elemento dentro de un documento. Un recurso también puede ser un objeto que no sea directamente asequible vía la web, por ejemplo un libro impreso, un recurso siempre tendrá un nombre (URI), Uniform Resource Identifier, *(Masinter, Berners-Lee, & Fielding, 2005)* que no es más que una cadena de caracteres corta que identifica inequívocamente un recurso.

Una propiedad es un aspecto, característica, atributo, o relación usada para describir un recurso. Cada propiedad tiene un significado especial, define valores, tipos de recursos, y relaciones con otras propiedades.

Una sentencia es un recurso específico junto a una propiedad nombrada más el valor que esta propiedad agrega al recurso, estas tres partes individuales y las sentencias son llamadas respectivamente, sujeto, predicado, objeto.

WEB ONTOLOGY LANGUAGE.

OWL *(McGuinness & Van Harmelen, 2004)* lenguaje web ontológico, fue diseñado para ser usado por aplicaciones que necesiten procesar el contenido de la información en lugar de sólo presentar la información a los humanos. OWL facilita a las máquinas la interpretabilidad de los contenidos web, está soportado por XML, RDF, y RDF esquema que

le provee vocabulario adicional más allá de una semántica formal. OWL tiene tres sub lenguajes expresivos incrementalmente: OWL Lite, OWL DL y OWL Full.

OWL puede ser usado para explicitar representaciones de significado de términos en vocabularios y relaciones entre estos términos. Estas relaciones de términos y sus interrelaciones son llamadas una ontología. OWL tiene más facilidad de expresar significados y semántica que XML, RDF, y RDF-S, OWL va más allá de estos lenguajes con su habilidad para representar contenido interpretable por máquinas.

La Web Semántica es una visión del futuro de la web en el cual la información tiene significado explícito haciendo más fáciles a las máquinas su procesamiento e integración de la información disponible. La Web Semántica fue construida con la habilidad que XML tiene para definir etiquetas personalizadas y con RDF que brinda un mecanismo para representación de los datos. El primer nivel sobre RDF que es requerido para una Web Semántica es un lenguaje ontológico que pueda formalmente describir el significado de la terminología usada en un documento web.

v. AGRADECIMIENTOS

Agradezco a mi familia por su apoyo incondicional y confianza depositada en mí, permitiéndome seguir adelante.

Agradezco a mi asesor de tesis PhD Julio Cesar Chavarro Porras, sus consejos y conocimiento fueron indispensables para llevar a cabo esta tesis.

A todos los compañeros del grupo de investigación GIA, sus consejos ayudaron a orientar y encaminar esta tesis.

vi. RESUMEN

Este trabajo describe el diseño de un operador de matching ontológico aplicable en el modelo conceptual propuesto por Chavarro (Chavarro, 2012), un diseño de cuatro capas que permite la configuración del operador de diferentes tipos de técnicas tanto a nivel de elemento como a nivel de estructura. El objetivo de este diseño es la creación de matcher ontológico de alto nivel adaptables a las necesidades de las ontologías con las cuales opera.

Este trabajo presenta las primeras fases de construcción del operador y sus resultados preliminares.

1

1 INTRODUCCIÓN

1.1 PRESENTACIÓN

En los últimos 15 años la web ha crecido de una herramienta para un grupo de científicos en el CERN a un espacio global de información con más de 3.2 billones de usuarios y se ha convertido en una fuente de información incalculable, más de 975 millones de sitios web y más de 32.000 GB de datos transferidos (*Stats, 2016*), esta cantidad de información hace que los investigadores de diferentes disciplinas lo consideren como el repositorio de conocimiento más grande existente. La web se ha introducido en todos los aspectos de la sociedad, desde una reserva en un hotel hasta un repositorio de información de un dominio particular. Con la llegada de la web 2.0

(O'Reilly, 2005) los usuarios son más participativos y juegan un papel importante en la generación de conocimiento, ya no como solo espectadores, sino como productores de información haciendo uso de todas estas herramientas que permiten leer y editar un sitio web, según cifras de Pingdom¹ en el 2015 se crearon más de 51 millones de sitios web y se realizaron más de 1.4 trillones de búsquedas en Google², sin embargo hay otra idea polémica para el futuro de la web, propuesta por Tim Berners-Lee, la Web Semántica (Berners-Lee, 2000), esta se basa en la idea de añadir datos a los datos (metadatos) semánticos y ontológicos, toda esta información adicional ayuda a describir el contenido, el significado y la relación entre los datos.

Las máquinas no pueden entender la información de la manera como los humanos lo hacemos, la Web Semántica no tiene que ver con entender los datos, sino como ponerlos en la web para que estén accesibles y podamos utilizarlos (Powell, 2007) , existe una cantidad enorme de

¹ <https://www.pingdom.com/>

² <https://www.google.com/>

información es por eso que algunos hablan de una web de datos «web profunda» que sólo es accesible a través de sitios web que tienen bases de datos detrás, entonces, la Web Semántica tiene que ver con subir información a la web de manera que pueda exponerse la web profunda, si la web de hipertexto nos brinda un espacio de información común, global, la Web Semántica nos brinda una trama de datos, permitiendo que la máquina analice el contenido y pueda interpretarla (Powell, 2007).

La Web Semántica tiene como objetivo fundamental que la información contenida en las páginas web, no sólo las entiendan las personas, sino que también puedan ser usadas como fuente de conocimiento por sistemas informáticos.

La forma como hacemos esto es a través de ontologías; las cuales definen los términos utilizados para describir y representar un área de conocimiento, y son utilizadas por personas, bases de datos y aplicaciones, que necesitan compartir información específica sobre un determinado tema o dominio.

Las ontologías computacionales son un medio para modelar formalmente la estructura de un sistema, es decir, las

entidades relevantes y las relaciones que surgen de su observación, y que son útiles para nuestros propósitos.

(Staab & Studer, 2009)

Una ontología es una entidad computacional, y no ha de ser considerada como una entidad natural que se descubre, sino como recurso artificial que se crea *(Mahesh, 1996)*

Una ontología ha de entenderse como un entendimiento común y compartido de un dominio, que puede comunicarse entre expertos de un dominio y sistemas computacionales. Ésta última característica, el hecho de que puedan compartirse y reutilizarse en aplicaciones diferentes, explica en parte el gran interés suscitado en los últimos años en la creación e integración de ontología.

(Gangemi, Pisanelli, & Steve, 1998)

Los conceptos de un dominio, están sujetos al cambio, a la evolución. Las ontologías, por tanto, son entes dinámicos, reflejan cambios que provienen del dominio que modelan, de la representación de los conceptos o de la implementación de la ontología en un lenguaje computacional.

La gestión de cambio es entendida como el proceso que permite representar, implementar, mantener y propagar la actualización de una ontología, en contextos abiertos y

distribuidos; este proceso incluye la creación de la ontología y las tareas que involucran múltiples ontologías que corresponden a dominios similares como son la reutilización, mezcla, alineación, e integración de ontologías. Todo este proceso se referencia como gestión del ciclo de vida ontológico, pero todas estas tareas son tareas complejas, producto de otras tareas previas, entre esas tareas se encuentra el emparejado (matching) ontológico.

1.2 DEFINICIÓN DEL PROBLEMA

El presente trabajo de creación de un «operador» para matching Ontológico en el modelo conceptual, esta soportado por la propuesta de creación del modelo conceptual para el cambio ontológico, este modelo está soportado en un modelo teórico que extiende el metamodelo de definición de ontologías (ODM), con un metamodelo de operaciones de cambio, y un modelo tecnológico para el desarrollo de un marco de referencia.

El metamodelo de operaciones de cambio, generaliza las operaciones que pueden ser aplicadas a una ontología y establece las correspondencias entre las operaciones y los metamodelos que representan los lenguajes de implementación de ontologías.

En el emparejamiento, se trata de buscar dos entidades correspondientes. Estas no tienen que ser necesariamente las mismas. Un emparejamiento puede ser, por ejemplo, en términos de un cerrojo y la llave que le sirva. Un grado de similitud sobre una dimensión especificada, por ejemplo, el patrón de cerrojo/llave. La combinación permite varias relaciones al mismo tiempo, mientras que la comparación implica un tipo específico de relación. Cualquier esquema de emparejamiento o algoritmo de emparejamiento de ontologías puede ser utilizado para implementar el operador Match. El emparejamiento concuerda con la definición de alineamiento general, sin embargo, donde hay una relación fija entre las entidades alineadas, expresa un tipo de emparejamiento (*Ehrig, 2007*).

En este proyecto se plantea la construcción de un operador (matcher) configurable, para hacer matching ontológico en la capa conceptual, que es independiente de su lenguaje de implementación y del modelo de representación del conocimiento subyacente, ya que todos los modelos de alineación de ontologías son dependientes del lenguaje de implementación de la ontología.

Se propone generar un operador (matcher) configurable para hacer *matching* ontológico en la capa conceptual, independiente del lenguaje de implementación y del modelo de representación del conocimiento.

1.3 MOTIVACIÓN

Al navegar por la web la información está representada generalmente como paginas HTML enriquecidas con CSS y en otros casos en XML, para acceder a esta información el usuario debe seguir una serie de hipervínculos que lo llevan de un sitio a otro, o puede realizar búsquedas por palabras claves, en ambos casos es el usuario quien decide qué información es relevante o no, sobre el tema buscado, este método de navegación presenta limitaciones y se convierte en una tarea pesada, pues la cantidad de información devuelta por los buscadores resulta inmanejable y el usuario termina perdiéndose, pues generalmente es más información de la que el usuario puede manejar.

Un primer problema a resolver cuando se desea extraer información, es el hecho de poder trabajar sobre un vocabulario común para referirse a los diferentes conceptos del dominio, es decir, asociarle un significado inequívoco a cada concepto. Este problema se resuelve definiendo

una ontología sobre el dominio que se desee trabajar (Fernández, 2004).

Como está definido en (Pérez, 2005) una ontología es una entendimiento común y compartido de un dominio, que puede compartirse entre científicos y sistemas computacionales, entre personas y sistemas heterogéneos, es decir, es una base de conocimiento que describe conceptos de un dominio particular, y de cómo los conceptos están relacionados.

La ontología está sujeta a cambios que pueden provenir del dominio, de la conceptualización, o de su especificación. (Pingdom, 2016), (José A. Senso & De la Rosa Piñero, 2003), (Howe, 1993), (J. A. Senso & Piñero, 2003). El efecto del cambio se puede extender a sus datos, o a las aplicaciones que usan la ontología. Se denomina gestión de cambio, al proceso que permite representar, implementar, mantener y propagar la actualización de una ontología.

1.4 ALCANCES Y CONTRIBUCIONES

A continuación, se enumeran las restricciones y alcances que se definieron en el proceso de esta investigación.

- Análisis de los métodos y metodologías del *matching* ontológico.

- Análisis de la interoperabilidad semántica.
- Construcción de una clasificación de técnicas de *matching* ontológico en el modelo conceptual. Este modelo conceptual está formulado en la tesis doctoral del ingeniero Julio César Chavarro Porras «Marco De Referencia Para La Gestión Del Cambio En Ontologías Distribuidas Para La Web Semántica: Búsqueda, Reutilización, Mezcla E Integración». (Chavarro, 2012)
- Evaluación de los principales sistemas de *matching*.

A nivel conceptual:

- Validar el modelo que se tomó como referencia comprobando que permite representar una ontología y nos abstrae los problemas de las capas léxica y sintáctica, y nos permite enfocarnos en las capas semántica y pragmática.

A nivel práctico:

- La realización de un análisis del desempeño de algunos métodos de *Matching*.

1.5 HIPÓTESIS Y METODOLOGÍA DE INVESTIGACIÓN

Motivada por las secciones anteriores, en esta sección se abordan algunas hipótesis sobre el planteamiento del problema mencionado y se describe el método usado para abordar el problema. Las siguientes suposiciones son consideradas:

H.1.. Es posible separar las operaciones de cambio ontológico en los niveles semántico y pragmático de los niveles léxico y sintáctico. Esta hipótesis es heredada del trabajo de *(Chavarro, 2012)*

H.2. La combinación de diferentes técnicas permite tener mejores resultados que técnicas individuales

H.3. Es posible el diseño de un operador de emparejado ontológico en el modelo conceptual propuesto por *(Chavarro, 2012)*

Para el logro de este trabajo de investigación se realizará una revisión crítica del estado del arte en matching ontológico, revisando los diferentes sistemas, su configuración, diseño tecnológico y enfoque.

Los sistemas a evaluar se tomarán de las últimas participaciones en OAEI³ (Ontology Alignment Evaluation Initiative)

El modelo propuesto se validará frente al trabajo desarrollado en «Marco de Referencia para la Gestión del Cambio en Ontologías Basados en Modelos Conceptuales» (Chavarro, 2012) y se realizarán pruebas unitarias a las técnicas con seleccionadas, aplicadas a ontologías conocidas o tomadas de OAEI.

1.6 ESTRUCTURA DE LA TESIS

A continuación, se define la estructura de este documento.

En el **capítulo uno** se presenta una breve introducción a la Web Semántica y a las ontologías y al problema del cambio ontológico, se define el problema al que se enfoca esta tesis y se muestran los alcances y contribuciones del trabajo de investigación.

³ <http://oaei.ontologymatching.org/>

En el **capítulo dos** se definen los objetivos principales y secundarios que se esperan cumplir al desarrollar esta tesis.

En el **capítulo tres** se definen conceptos y bases teóricas, el concepto de ontología, las propiedades, características y clasificaciones, se presentan algunas técnicas de *matching* y se hace una explicación del modelo conceptual.

En el **capítulo cuatro** se muestra el diseño del operador propuesto desde el punto de vista arquitectural, como desde el punto de vista tecnológico.

En el **capítulo cinco** se muestran las conclusiones de esta tesis referente al diseño del operador de *matching* ontológico en el modelo conceptual, se presentan algunas posibles líneas como trabajo futuro de este proyecto.

En los **anexos** podemos encontrar un análisis de algunas de los principales sistemas de *matching*, así como de revisión de algunas de las principales técnicas de *matching* a nivel de elementos.

2

2 OBJETIVOS

El objetivo principal de la tesis es el diseño de un operador, configurable por el usuario, para realizar el proceso de *matching* ontológico en los niveles de representación semántico y semiótico.

2.1. OBJETIVOS ESPECÍFICOS

Los objetivos específicos son:

El objetivo principal nos marca una línea general de investigación y de allí se desglosan los siguientes objetivos específicos.

- Realizar un análisis crítico del estado del arte en el campo de la alineación ontológica.

- Realizar un estudio comparativo de los algoritmos de *matching* de ontologías en el estado del arte actual.
- Realizar la adaptación de algoritmos de *matching* al modelo conceptual.
- Implementar algoritmos de *matching* ontológicos aplicables en los niveles de representación semántico y semiótico.
- Proponer un matcher para alineación de ontologías en el modelo conceptual.

3

3 FUNDAMENTOS TEÓRICOS, TECNOLOGÍAS Y ESTADO DEL ARTE

En este capítulo se presenta una revisión de los fundamentos teóricos, tecnologías y el estado del arte. Se adopta una definición de ontología para el resto de la tesis, se muestra el problema del cambio ontológico y se realiza una explicación del modelo conceptual.

Con la comprensión de las técnicas de *matching* y el modelo conceptual nos ubicamos en el terreno objeto de esta tesis, el cual es, la definición de un operador basado en las técnicas existentes pero aplicado al modelo conceptual descrito en el trabajo «marco de referencia para la gestión del cambio ontológico» ([Chavarro, 2012](#)).

3.1 ONTOLOGÍA

3.1.1.CONCEPTO DE ONTOLOGIA

La palabra ontología se deriva del griego ontos (estudio del ser) y logos (palabra). Filosóficamente, ontología es la ciencia de qué es, es una explicación sistemática de la Existencia, de los tipos de estructuras, categorías de objetos, propiedades, eventos, procesos y relaciones en cada área de la realidad. En Filosofía, ontología es la teoría de «la naturaleza de las cosas o los tipos de existencia». Los filósofos griegos Sócrates y Aristóteles fueron los primeros en desarrollar los fundamentos de las ontologías. Sócrates introdujo la noción de ideas abstractas, una jerarquía entre ellas, y relaciones clase - instancia. Aristóteles agregó asociaciones lógicas. El resultado es un modelo bien estructurado capaz de describir el mundo real.

En la historia moderna, los primeros artículos que resumen a la ontología como disciplina filosófica fueron publicados alrededor de 1960 (*Strawson & Bubner, 1975*). En la rama de Inteligencia Artificial y los investigadores web se ha adoptado el término «ontología» para sus necesidades. Actualmente, hay diferentes definiciones en la literatura, de que cosa debe ser una ontología.

Una ontología puede ser vista como un conjunto de afirmaciones con el propósito de modelar un dominio en particular. Generalmente la ontología define un vocabulario usado por una aplicación específica, en las ciencias de la computación existen diferentes modelos de datos y modelos conceptuales que puede ser pensados como una ontología: esquemas de bases de datos, modelos UML, esquemas XML, cada uno de estos ejemplos tienen diferentes grados de formalidad, pero en el nivel más alto está la ontología, que tiene un mayor nivel de expresividad y una semántica mejor definida.

Las ontologías representan los conocimientos de forma genérica y consensuada para todo usuario de ellas. Sus definiciones pueden ser reutilizadas y compartidas en distintos dominios y por diferentes aplicaciones. Proporcionan un vocabulario unificado, consistente, y coherente, expresado en lenguajes procesables por una computadora y con distinto grado de formalidad, que permite que una aplicación pueda dialogar con otra porque las dos entienden lo mismo, aunque los términos empleados sean distintos. Permiten el intercambio de información y conocimientos

entre aplicaciones heterogéneas desde una perspectiva semántica y sintáctica. (Gómez, 2008)

Para este trabajo se entenderá una ontología como: «una especificación formal y explícita de una conceptualización compartida» (Studer, Benjamins, & Fensel, 1998), por especificación formal hace referencia a los conceptos y relaciones del modelo abstracto especificados en un lenguaje estructurado y comprensible por las máquinas, y por conceptualización compartida hacemos referencia al contexto particular en el cual se definió la ontología, al mundo tal y como lo ve el experto que define la ontología.

La característica distintiva de una ontología en comparación con otro modelo de representación del conocimiento, es la existencia de un modelo semántico teórico: las ontologías son teorías lógicas.

Las ontologías son expresadas en un lenguaje ontológico, existen diferentes lenguajes de representación, pero la mayoría de estos lenguajes comparten el mismo tipo de entidades, con diferentes nombres pero con significados comparables, ejemplos de estos lenguajes son: RDF (Lassila & Swick, 1998) y OWL (McGuinness & Van Harmelen, 2004).

3.1.2.DEFINICION DE ONTOLOGIA

Aunque se han propuesto diferentes definiciones de ontología, en este trabajo se parte de la aportada por Gruber». (Gruber, 1995) y complementada por (Studer et al., 1998) «Una ontología es una especificación explícita y formal de una conceptualización» Una conceptualización se refiere a un modelo abstracto de algún fenómeno del mundo identificando los conceptos relevantes de ese fenómeno, (Studer et al., 1998).

En resumen, vamos a considerar una ontología con la siguiente definición tomada de Euzenat y Shvaiko (Euzenat & Shvaiko, 2007), y es la que se usará en esta tesis.

Una ontología es una tupla $O = \{C, I, R, T, V, \leq, \perp, \in, =\}$, tal que

- C es un conjunto de clases,
 - I es un conjunto de individuos,
 - R es un conjunto de relaciones,
 - T es un conjunto de tipos de datos,
 - V Es un conjunto de valores
- $(C, I, R, T, V \text{ disjuncto por pares}),$
- \leq es una relación en $(C \times C) \cup (R \times R) \cup (T \times T)$ llamada especialización,

- \perp es una relación en $(C \times C) \cup (R \times R) \cup (T \times T)$ llamada exclusión,
- \in es una relación sobre $(I \times C) \cup (V \times T)$ llamada instanciación
- $=$ es una relación sobre $I \times P \times (I \cup V)$ llamada asignación.

INTERPRETACIÓN

Dada una ontología $O = \{C, I, R, T, V, \leq, \perp, \in, =\}$ una interpretación de O es un par $\{I, D\}$, donde D es llamado el dominio de interpretación e I es una función llamada función de interpretación, tal que:

$$\forall c \in C, I(c) \subseteq D;$$

$$\forall r \in R, I(r) \subseteq D \times (D \cup V);$$

$$\forall i \in I, I(i) \in D;$$

$$\forall t \in T, I(t) \subseteq V;$$

$$\forall v \in V, I(v) \in V;$$

Una afirmación expresada en un lenguaje ontológico se dice que es satisfecha por una interpretación, si la interpretación es coherente con esta afirmación.

SATISFACIBILIDAD

Dada una ontología $O = \{C, I, R, T, V, \leq, \perp, \in, =\}$, una fórmula δ , el cual es satisfecho por la interpretación $\{I, D\}$ de O denotada como $I \models \delta$, es definida:

$$I \models c \leq c' \text{ Si y solo si } I(c) \subseteq I(c')$$

$$I \models r \leq r' \text{ Si y solo si } I(r) \subseteq I(r')$$

$$I \models t \leq t' \text{ Si y solo si } I(t) \subseteq I(t')$$

$$I \models c \perp c' \text{ Si y solo si } I(c) \cap I(c') = \emptyset$$

$$I \models r \perp r' \text{ Si y solo si } I(r) \cap I(r') = \emptyset$$

$$I \models t \perp t' \text{ Si y solo si } I(t) \cap I(t') = \emptyset$$

$$I \models i \in c \text{ Si y solo si } I(i) \in I(c)$$

$$I \models v \in t \text{ Si y solo si } I(v) \in I(t)$$

$$I \models i.r = i' \text{ Si y solo si } \{I(i), I(i')\} \in I(t)$$

$$I \models i.r = v \text{ Si y solo si } \{I(i), I(v)\} \in I(r)$$

Las formulas ontológicas pueden contener más de estas afirmaciones. Una ontología es un conjunto de afirmaciones que seleccionan un conjunto de interpretaciones que la satisfacen. Esta interpretación es llamado modelo. Ello constituye la posible interpretación de una ontología.

MODELO

Dada una ontología o , un modelo de o es una interpretación $m = \{I, D\}$ de o , el cual satisface todas las afirmaciones en o :

$$\forall \delta \in o, m \models \delta$$

El conjunto de modelos de una ontología o es denotado como $M(o)$.

CONSECUENCIA

Dada una formula ontológica δ , δ es una consecuencia de una ontología o , si y solo si, es satisfecha por todos los modelos de o . Esto es denotado como $o \models \delta$.

Dado un modelo m , vamos a denotar como $m(e)$ la aplicación de la función de implementación del modelo en alguna entidad e .

3.1.3.CLASIFICACION DE ONTOLOGIAS

Dependiendo del punto de vista de la gestión del conocimiento que se adopte, se pueden diferenciar múltiples tipos de Ontologías.

Ontologías finas y gruesas: una ontología fina consta de un rico conjunto de axiomas y relaciones conceptuales de relevancia, que permiten definir con bastante precisión los conceptos a los que se refiere. Por su parte, una ontología

gruesa consiste en un conjunto reducido de axiomas formulados en un lenguaje de limitada capacidad expresiva. Las ontologías finas, a veces son consideradas ontología de referencia por su precisión y nivel de detalle, mientras que las ontologías gruesas se conocen también como ontologías compatibles. (Guarino, 1998)

Otra clasificación propuesta por (Lassila & McGuinness, 2001) es en función de su estructura interna, bajo esta clasificación, los distintos tipos de ontologías pueden visualizarse dispuestos en un espectro o rango lineal como el representado en la Figura 1.

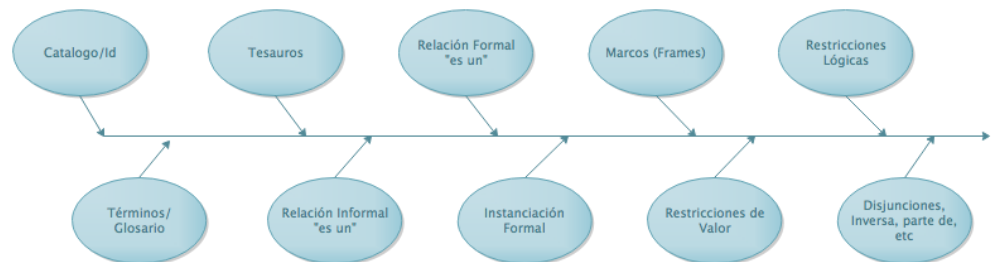


Figura 1. Espectro de ontologías (grafica según (Lassila & McGuinness, 2001))

Así, las ontologías se clasifican según:

- Vocabularios controlados, esto es una lista finita de términos, como puede ser un catálogo.

- Glosarios, es una lista de términos junto con su significado. El significado de los términos se especifica por medio de enunciadas en lenguaje natural.
- Tesauros, provee información adicional acerca de las relaciones entre términos, como las de sinonimia.
- Jerarquías de términos informales, hace referencia a ciertas ontologías que se pueden encontrar en la web y en las relaciones «es un» no son definidas en sentido estricto.
- Jerarquías de subclases formales o estrictas, las jerarquías de subclases estrictas se utilizan para explorar las propiedades de herencia.
- Jerarquía de subclases formales y relaciones, estas ontologías incluyen también relaciones entre conceptos para describir un dominio concreto.
- Marcos (Frames), en estas ontologías las instancias de una clase pueden contener información acerca de sus propiedades. Esas propiedades serán heredadas por las instancias de las subclases de dicha clase.

- Restricciones de valor, en estas ontologías se pueden definir restricciones acerca de qué valores pueden tomar las propiedades de un objeto de una clase.
- Restricciones lógicas con carácter general, estas son las ontologías más expresivas. Con ellas, los ingenieros ontológicos pueden especificar relaciones entre términos en lógica de primer orden.

Conforme se avanza en la línea del espectro, se va avanzando a su vez en expresividad y poder semántico.

Otra clasificación según [\(Guarino, 1998\)](#), es basada en el nivel de generalidad o en el propósito de la ontología, existen 4 tipos o clases de ontologías, representados en la **Figura 2**

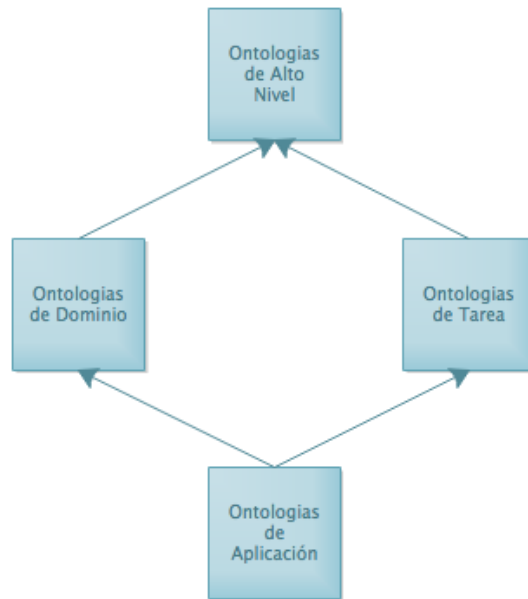


Figura 2. Tipos de ontología según su nivel de generalidad (grafica según *(Guarino, 1998)*)

- Las ontologías de alto nivel describen los conceptos generales como el espacio, tiempo, materia, objeto, evento, acción, los cuales son independientes de un problema o dominio en particular.
- Las ontologías de dominio y de tareas describen respectivamente el vocabulario relacionado a un dominio genérico, por ejemplo: medicina, o una tarea o actividad genérica, como: diagnósticos, especializando los términos introducidos en la ontología de alto nivel.
- Las ontologías de tareas: describen el vocabulario relativo a una tarea genérica o actividad, como, por

ejemplo, diagnosticar, planificar, vender, etc. Para ello, al igual que ocurría con las ontologías de dominio, se especializan los términos introducidos en ontologías de alto nivel.

- Las ontologías de aplicación describen conceptos dependiendo de un dominio y de una tarea en particular, la cual es una especialización de ambas ontologías relacionadas (ontología de dominio y ontología de tarea). Estos conceptos usualmente corresponden a un papel jugado por las entidades de dominio mientras realizan una actividad).

Otras clasificaciones son:

Según el grado o nivel de abstracción y razonamiento lógico que permitan, por ejemplo: ontologías descriptivas, que incluyen taxonomías de conceptos, relaciones entre conceptos, pero no permiten inferencias lógicas y ontologías lógicas, las que permiten inferencias lógicas mediante la utilización de una serie de componentes como la inclusión de axiomas.

3.2 CAMBIO ONTOLÓGICO

Una ontología está formada por entidades, una taxonomía de conceptos y por un conjunto de axiomas o reglas de inferencia mediante las cuales se pueden inferir conocimiento nuevo a partir del conocimiento que subyace la representación, las entidades toman valores de acuerdo a un dominio determinado, pero este dominio evoluciona, modificando sus límites dado que el uso de las ontologías es en la Web Semántica, un ente abierto y distribuido y en constante evolución.

La ontología pronto dejará de ser pertinente y necesitará actualizarse para adaptarse a su entorno, la ontología está sujeta a cambio en su dominio, a la conceptualización o a su especificación y estos cambios afectan a sus datos, a las aplicaciones que hace uso de ella y a otras ontologías que hacen uso de esta o que sus dominios se solapan o comparten.

Durante el ciclo de vida de una ontología es necesario adaptarse a nuevos requerimientos, uno de estos es unirse para compartir o crear una nueva ontología a partir de las existentes, estos procesos pueden ser mezcla, integración, fusión de ontologías. Para gestionar este cambio se han

descrito varias perspectivas, las dos perspectivas más utilizadas son el control de versiones (*M. Klein, 2004*), y la evolución ontológica (*Stojanovic, Maedche, Motik, & Stojanovic, 2002*); estas perspectivas han sido implementadas en los Framework Protegé⁴, KAÓN⁵, WebODE⁶.

Según estas perspectiva de cambio, los ingenieros ontológicos necesitan guardar las versiones de las ontologías, para comparar campos o para dar soporte a aplicaciones que no se han actualizado a la nueva ontología, pero cuando una nueva ontología es producida por un proceso de fusión, mezcla o integración, dado el tamaño de la ontología, los ingenieros ontológicos no pueden saber cuáles segmentos se modificaron y cuáles no, es aquí donde ayuda el proceso de *matching* ontológico, encontrando esa correspondencia entre dos versiones de una misma ontología.

La ontología está sujeta a cambios, diversos autores han descrito las causas, la gestión y los efectos del cambio.

⁴ <http://protege.stanford.edu/>

⁵ <http://kaon.semanticweb.org/>

⁶ <http://webode.dia.fi.upm.es/WebODEWeb/index.htm>

((M. Klein & Fensel, 2001); (M Klein, Fensel, Kiryakov, & Ognyanov, 2002); (M. Klein, 2004); (Noy & Klein, 2002)), definen tres tipos de fuentes de cambio: Cambios en el **dominio** que se representa, cambios en la **conceptualización**, los cuales pueden provenir de un cambio en la perspectiva de uso, o de un cambio en la vista del universo del discurso, y cambios en la **especificación**, es decir cambios en el lenguaje en que se implementa la ontología. El efecto del cambio se puede extender a sus datos, o a las aplicaciones que usan la ontología.

En el proceso de gestión de cambio, se deben realizar tareas que se consideran complejas, estas son: mezcla, fusión, integración y segmentación. Para llevar a cabo estas tareas es necesario realizar un proceso de *matching* ontológico, del cual se obtiene como resultado una alineación o *mapping*, este es un proceso común a todas las tareas que involucran múltiples ontologías.

El proceso de emparejar ontologías (*matching*) y la alineación o *mapping* obtenido, corresponde a una fase de las tareas de mezcla, fusión, integración y segmentación, e incluso puede ser usado en tareas como procesamiento de consultas y edición ontológica.

La meta del emparejamiento ontológico es reducir la heterogeneidad y la interoperabilidad entre ontologías, como podemos ver en la **sección 3.2.1** tomado de (*Flouris, Plexousakis, & Antoniou, 2006*) se resumen varios subcampos en el cambio ontológico.

Han habido diferentes clasificaciones de heterogeneidad, algunos enfocados en los desacoples (*M. Klein & Fensel, 2001*) y otros en los niveles de interoperabilidad (*Euzenat, 2001*), pero para que esto sea posible hay que superar cuatro principales problemas al momento de buscar la interoperabilidad entre ontologías. (*Euzenat et al., 2010*).

Una es la **heterogeneidad sintáctica** se refiere cuando la información de diferentes fuentes es almacenada usando métodos de codificación diferente entre sí, por ejemplo, cuando tenemos una ontología en OWL y otra en RDF. Esto también ocurre cuando dos ontologías son modeladas usando diferentes formalismos del conocimiento.

Otra es la **heterogeneidad semántica** sucede cuando para representar la información se utilizan términos diferentes para representar el significado de los mismos datos. Por ejemplo, los términos «PROGRAMAS DE ORDENADOR» y «SOFTWARE PARA COMPUTADORAS» tienen el mismo

significado, pero son expresiones distintas, esto también puede pasar debido al uso de diferentes axiomas para definir un concepto.

En el contexto de diferencias conceptuales se identifican tres importantes razones por la cual ocurre esto

- Diferencia en **cobertura**: ocurre cuando dos ontologías describen de manera diferente regiones del modelo al mismo nivel de detalle y desde una única perspectiva, posible superposición.
- Diferencia en **granularidad**: ocurre cuando dos ontologías describen la misma región del modelo desde una misma perspectiva, pero en diferente nivel de detalle.
- Diferencia en **perspectiva**: también llamada diferencia en alcance, ocurre cuando dos ontologías describen la misma región del universo, al mismo nivel de detalle, pero desde diferentes perspectivas.

Otra es la **Heterogeneidad estructural**: es determinada a través de las ocurrencias de las propiedades incorporadas usadas en las dos ontologías a ser alineadas, es decir, vocabulario para definir propiedades, el cual es incorporado y usados por los lenguajes OWL y RDF,

Y por último **Heterogeneidad en modelos**: diferencias en los modelos subyacentes (bases de datos, ontologías) o su representación (modelo relacional, orientado a objetos, lenguajes de ontologías RDF, OWL).

3.2.1.RESUMEN DE CAMBIOS ONTOLOGICOS

MAPEO ONTOLÓGICO

Propósito:	Resolución de heterogeneidad e interoperabilidad.
Entrada:	Dos ontologías heterogéneas.
Salida:	Un mapeo entre entidades.
Propiedades:	La salida identifica la relación entre las entidades.

Tabla 1. Mapeo ontológico

TRANSFORMACIÓN ONTOLÓGICA

Propósito:	Resolución de heterogeneidad e interoperabilidad.
Entrada:	Dos ontologías heterogéneas.
Salida:	Mapeo entre entidades y axiomas.

Propiedades:	La salida identifica entidades y axiomas relacionados.
--------------	--

Tabla 2. Transformación Ontológica

ALINEACIÓN ONTOLÓGICA

Propósito:	Resolución de heterogeneidad e interoperabilidad.
Entrada:	Dos ontologías heterogéneas.
Salida:	Una relación entre entidades.
Propiedades:	La salida identifica las relaciones entre entidades.

Tabla 3. Alineación Ontológica

ARTICULACIÓN ONTOLÓGICA

Propósito:	Resolución de heterogeneidad, interoperabilidad.
Entrada:	Dos ontologías heterogéneas.

Salida:	Una ontología intermedia y mapeo entre entidades de la ontología intermedia con cada ontología fuente.
Propiedades:	La salida es equivalente a una relación identificando entidades relacionadas.

Tabla 4. Articulación Ontológica

TRASLACIÓN ONTOLÓGICA (PRIMER ENFOQUE)

Propósito:	Usar un diferente lenguaje de representación.
Entrada:	Una ontología y un lenguaje de representación destino.
Salida:	Una ontología representada en el lenguaje destino.
Propiedades:	Produce una ontología equivalente, de ser posible.

Tabla 5. Traslación Ontológica (primer enfoque)

TRASLACIÓN ONTOLÓGICA (SEGUNDO ENFOQUE)

Propósito:	Implementar un mapeo
Entrada:	Una ontología y un mapeo.
Salida:	Una ontología
Propiedades:	Implementa un mapeo.

Tabla 6. Traslación Ontológica (segundo enfoque)

EVOLUCIÓN ONTOLÓGICA

Propósito:	Aplicar cambios de dominio o de conceptualización.
Entrada:	Una ontología y una operación de cambio.
Salida	Una ontología.

Propiedades:	Implementa cambios en la ontología fuente.
--------------	--

Tabla 7. Evolución Ontológica

CONTROL DE VERSIONES DE ONTOLOGÍAS

Propósito:	Acceso transparente a diferentes versiones.
Entrada:	Diferentes versiones de una ontología.
Salida:	Un sistema de control de versiones.
Propiedades:	Identificar versiones, acceso transparente a las versiones, determinación de compatibilidad.

Tabla 8. Control de Versiones de ontologías

INTEGRACIÓN ONTOLÓGICA

Propósito:	Fusionar ontologías de dominio similar.
Entrada:	Dos ontologías, de dominio similar.
Salida:	Una ontología.

Propiedades:	Fusionar conocimiento para cubrir un borde de dominio.
--------------	--

Tabla 9. Integración Ontológica

MIGRACIÓN ONTOLÓGICA

Propósito:	Fusionar ontologías de dominios similares.
Entrada:	Dos ontologías que cubren dominios idénticos.
Salida:	Una ontología.
Propiedades:	Fusionar conocimiento para describir un dominio más preciso.

Tabla 10. Migración Ontológica

3.3 REPRESENTACIÓN DE UNA ALINEACIÓN

La alineación (*Alignment*), es el conjunto de correspondencias entre dos o más ontologías, la alineación es la salida de un proceso de *matching*.

Podemos definir la alineación de la siguiente forma:

Dadas dos ontologías o y o' , una alineación entre o y o' , es el conjunto de correspondencias ($4 - tuples$): $\{e, e', r, n\}$ con:

- $e \in o$ y $e' \in o'$ siendo las dos entidades emparejadas
- r es una relación entre e y e'
- n expresa en nivel de confianza $[0..1]$ en esta correspondencia

En algunos lenguajes la alineación se expresa haciendo uso de XML:

```
<uri_onto1 id='1'>Acapulco </uri1_onto1>
<uri_onto2 id='2'> Cancun</uri2_onto2>
<CorrespondenciasFuertes>
  <Pareja1>
    <entidad1 URI=Acapulco.hotel id='1'>
    <entidad2 URI=Cancun.hotel id='2'>
    <SimilitudGlobal>0.90</SimilitudGlobal>
    <relacion> "Correspondencia fuerte" </relacion>
    <entidad1 URI=Acapulco.motel id='1'>
    <entidad2 URI=Cancun. motel id='2'>
    <SimilitudGlobal>1.0</SimilitudGlobal>
    <relacion> "Correspondencia fuerte" </relacion>
  </Pareja1>
</CorrespondenciasFuertes>
```

3.4 **MATCHING ONTOLÓGICO**

El *matching* ontológico es el proceso de encontrar relaciones o correspondencias entre entidades de ontologías diferentes (Euzenat & Shvaiko, 2007), la correspondencia es la relación de pertenencia, o supuesta pertenencia según un algoritmo de *matching* particular entre entidades de diferentes ontologías. Estas entidades pueden ser clases, individuos, relaciones.

El emparejado ontológico es un proceso necesario dado que la Web Semántica es un sistema abierto y distribuido, donde muchas aplicaciones están presentes, muchos individuos interactúan allí, con diferentes intereses y en distinto grado de detalle, con distintas herramientas y diferentes grados de conocimiento. Es por esta razón que surgen varias clases de heterogeneidades, el *matching* ontológico entra allí como una ayuda a encontrar como estas diferentes ontologías se complementan, se solapan o se superponen aun cuando el dominio de estas parezca ser diferente.

La meta del *matching* ontológico es reducir la heterogeneidad entre términos, La heterogeneidad no se encuentra únicamente en las diferencias entre los objetivos de las

aplicaciones para la cual han sido diseñadas o en los formalismos expresión en la que las ontologías se han codificado (Euzenat & Shvaiko, 2007). El *matching* ontológico es una promesa de solución al problema de la heterogeneidad semántica, estas correspondencias pueden ser usadas para varias tareas, como lo es la migración ontológica, inferencia ontológica, traducción de datos, o para navegación por la Web Semántica, en consecuencia, el *matching* ontológico posibilita la interoperabilidad de conocimiento y datos.

3.4.1.EL PROCESO DE *MATCHING*

La operación *matching* (emparejado) determina la alineación A' para un par de ontologías o y o' . Existen otros parámetros que pueden extender la definición del proceso de *matching*:

- El uso de una alineación de entrada A .
- Los parámetros del *matching*, por ejemplo, peso, umbrales, etc.

- Recursos externos utilizados por el proceso de *matching*, por ejemplo, tesauros de especificación de dominio y conocimiento común.

El proceso de *matching* ontológico se puede definir de la siguiente forma:

El proceso de *matching* puede ser visto como una función f el cual, toma un par de ontologías a emparejar o y o' , una alineación de entrada A (opcional), un conjunto de parámetros p y un conjunto de recursos r , retorna una alineación A' entre estas ontologías:

$$A' = f(o, o', A, p, r)$$

Esta definición puede ser representada gráficamente como se muestra en la **Figura 3**

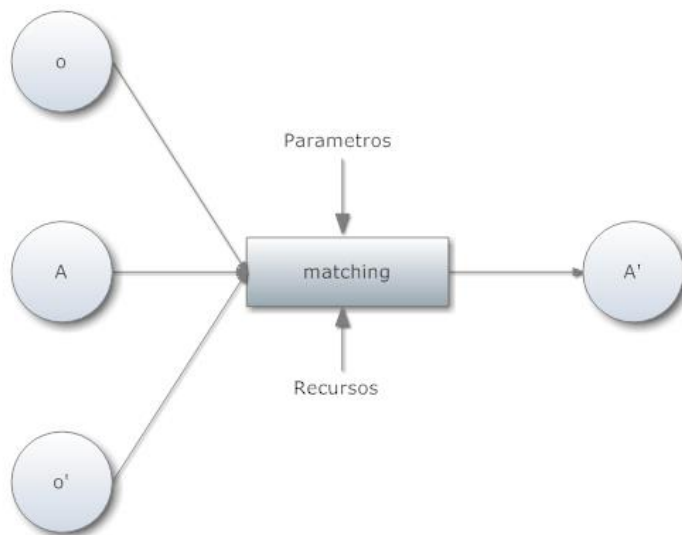


Figura 3. El proceso de *matching* (basado en (Euzenat & Shvaiko, 2007))

La definición anterior se puede extender a un proceso de *matching* múltiple donde o y o' se pueden considerar como un conjunto de ontologías de entrada.

$$A' = f(o_1, \dots, o_n, A, p, r)$$

3.4.2. CLASIFICACION DE LAS TECNICAS DE *MATCHING*

Luego de la definición del proceso del *matching* ontológico, vamos realizar una clasificación de técnicas que se han usado para resolver este problema. Esta clasificación está basada en el trabajo de Euzenat el cual a su vez se basó en los trabajos de (Rahm & Bernstein, 2001), (Wache et al., 2001), (Kalfoglou & Schorlemmer, 2003), que a su vez fueron basados en los trabajos previos de los cuales las mayores contribuciones fueron de (Larson, Navathe, & Elmasri, 1989), (Batini, Lenzerini, & Navathe, 1986), (Kashyap & Sheth, 1997), (Parent & Spaccapietra, 1998). En esos trabajos se enfoca el problema del *matching* desde diferentes perspectivas: inteligencia artificial, sistema de información, bases de datos.

CLASIFICACIÓN SEGÚN EL PROCESO DE MATCHING

Para la clasificación de las técnicas de *matching* se tuvo en cuenta:

- La entrada a los algoritmos.
- Las características del proceso de *matching*.
- La salida de los algoritmos.

La entrada: relacionado con el tipo de entrada con el cual el algoritmo opera, el algoritmo se puede clasificar dependiendo en el modelo de dato o conceptual en el cual la ontología es expresada, por ejemplo, modelo entidad, modelo orientado a objetos, RDF, OWL.

Otra posible clasificación dependiendo la entrada al algoritmo depende el tipo de información que el algoritmo utiliza, por ejemplo, a nivel de esquema o a nivel datos, a nivel de clase o nivel instancia.

El proceso: una clasificación del proceso de *matching* puede estar basada en las propiedades generales del proceso, éste depende si es aproximado o exacto. Un algoritmo exacto se enfoca en la precisión de la solución del problema, un algoritmo aproximado sacrifica la exactitud por el rendimiento. Otra clasificación puede estar basada

en la manera como el proceso interpreta los datos de entrada.

La salida: teniendo en cuenta la salida de los algoritmos las técnicas de *matching* se pueden clasificar según la forma como el resultado es producido y cómo esta salida es manipulada, algunos algoritmos utilizan la salida como un insumo de entrada para mejorar el proceso, en cambio otros generan alineaciones de una instancia a la vez.

CLASIFICACIÓN SEGÚN EL TIPO DE RELACIONES

Otra clasificación puede estar basada en el tipo de relaciones que el algoritmo pueda proporcionar, la mayoría de los sistemas se enfocan en equivalencias ($=$), mientras que otros se enfocan en suministrar resultados más expresivos, por ejemplo, subsunción (\subseteq), incompatibilidad (\perp), etc.

Según la clasificación realizada por Euzenat distinguió en dos grandes grupos sintácticos:

Granularidad/interpretación de entrada: esta clasificación está basada en la granularidad de los elementos o estructuras y en cómo las técnicas interpretan de manera general la información de entrada, a su vez se subdivide en:

a nivel de elemento o a nivel de estructura, esta clasificación depende de en cuales tipos de datos opera el algoritmo.

En las técnicas a nivel de elemento considera a las entidades ontológicas aisladamente de sus relaciones con las otras entidades. Por el contrario, en las técnicas a nivel de estructura considera a las entidades ontológicas o sus instancias para comparar las relaciones con otras entidades.

Tipo de entrada: esta clasificación está basada en el tipo de entrada la cual es usada por las técnicas de *matching* básicas.

Las técnicas básicas clasificadas en esta taxonomía son:

- **Técnicas basadas en cadenas:** en esta técnica se considera las cadenas como una secuencia de letras de un alfabeto, las funciones de distancia, mapea a las cadenas a un número real, donde el número indica el grado de similitud entre las cadenas, algunos sistemas tienen en cuenta los prefijos y sufijos para mejorar el proceso.

- **Técnicas basadas en lenguaje:** estas técnicas consideran los nombres como palabras en lenguaje natural, explotando las propiedades morfológicas de la cadena entrada.
- **Técnicas basadas en restricciones:** las técnicas basadas en restricciones, son algoritmos que utilizan las restricciones aplicadas para la definición de las entidades, como: tipos, cardinalidad, atributos, etc.
- **Técnicas basadas en recursos lingüísticos:** los recursos lingüísticos como los lexicones o tesauros son usadas para relacionar palabras. Estas técnicas utilizan las relaciones lingüísticas entre los términos, por ejemplo: sinónimos, antónimos, homónimos, etc.
- **Técnicas que rehúsan alineaciones:** estas técnicas utilizan la salida del proceso de *matching* como entrada a un nuevo proceso, son usadas especialmente con ontologías de gran tamaño, en este caso, primero dividen la ontología en ontologías más pequeñas a las cuales se le aplica el proceso de *matching* luego todas estas salidas son utilizadas para realizar el *matching* en la ontología original.

- **Técnicas que usan ontologías formales de alto nivel y dominio específico:** las ontologías de alto nivel son utilizadas como recursos externos de conocimiento común. Las ontologías de dominio específico pueden ser usadas como recursos externos o base de conocimiento. Estas ontologías se enfocan en dominios particulares y usan términos en un sentido relevante sólo en este dominio, por ejemplo, dominio médico.
- **Técnicas basadas en grafos:** estas técnicas consideran las ontologías de entrada como grafos etiquetados, generalmente, las comparaciones entre un par de nodos de las ontologías son basadas en su posición en el grafo, estas técnicas parten de la premisa: «si los nodos de dos ontologías son similares, sus vecinos también deben ser similares».
- **Técnicas basadas en taxonomía:** las técnicas basadas en taxonomía también utilizan algoritmos de grafos, pero sólo consideran la relación de especialización «es un».
- **Técnicas que usan repositorio de estructuras:** los repositorios de estructuras almacenan ontologías o

fragmentos de ellas y una media de similitud, el objetivo es identificar estructuras que sean suficientemente similares a las estructuras almacenadas, así se rehúsa alineaciones analizadas previamente.

- **Técnicas basadas en modelos:** también llamadas técnicas basadas en semántica, en el cual la entrada a sus algoritmos son interpretaciones semánticas, esta técnica se basa en la premisa de que, si dos entidades son las mismas, entonces, ellas también comparten la misma interpretación.
- **Técnicas estadísticas y de análisis de datos:** estas técnicas toman muestras representativas de una población para encontrar irregularidades o discrepancias, así ayuda a agrupar ítems o a computar distancias entre ellos.

3.4.3. ESTRATEGIAS DE *MATCHING*

En la **sección 3.4.1** se expuso el proceso de *matching* el cual de forma general es: una función f la cual, toma un par de ontologías a emparejar o y o' , una alineación de entrada A (opcional), un conjunto de parámetros p y un

conjunto de recursos r , retorna una alineación A' entre estas ontologías:

$$A' = f(o, o', A, p, r)$$

Pero para lograr un mayor grado de confianza en la alineación de salida se utilizan una serie de estrategias de combinación de diferentes métodos y técnicas, recordemos que las técnicas de *matching* las podemos clasificar en:

- Técnicas basadas en cadenas
- Técnicas basadas en lenguaje
- Técnicas basadas en restricciones
- Técnicas que usan recursos lingüísticos
- Técnicas que rehúsan alineaciones
- Técnicas que usan ontologías formales de alto nivel y dominio específico
- Técnicas basadas en grafos
- Técnicas basadas en taxonomía
- Técnicas que usan repositorio de estructuras.
- Técnicas basadas en modelos
- Técnicas estadísticas y de análisis de datos

Las agrupaciones de técnicas mostradas arriba son los bloques con los cuales construimos la solución, para esto debemos tener en cuenta los siguientes aspectos (Euzenat & Shvaiko, 2007)

- La combinación de resultados de métodos básicos para calcular la similitud entre entidades y organizar dicha combinación de diversas formas.
- El desarrollo de una estrategia para calcular similitudes a pesar de los ciclos y de la no linealidad en las restricciones que rigen las similitudes.
- Aplicar los métodos y parámetros más adecuados según los datos.
- Usar métodos probabilísticos para combinar *matcher* o para identificar correspondencias faltantes.
- Vincular a los usuarios en el proceso.
- Usar las alineaciones producto de la disimilitud resultante.

3.4.4. ESTRATEGIAS DE COMBINACION DE TECNICAS

SECUENCIAL

Ejecuta técnicas en secuencia, el resultado de la ejecución de la primera es la entrada para la ejecución de la segunda. A menudo, Se usan técnicas de nivel de estructura para mejorar las alineaciones creadas. Por ejemplo, realiza primero la concordancia basada en cadenas y utiliza candidatos de equivalencia como anclas (límites de trayectoria) para la concordancia de la trayectoria siguiente.

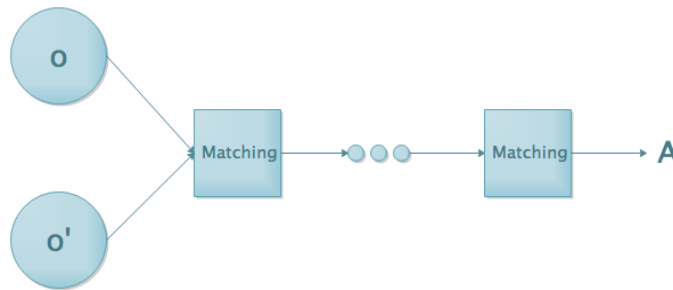


Figura 4. Estructura secuencial de *matching*

EN PARALELO

Un matcher con esta estructura utiliza simultáneamente diferentes técnicas básicas a las mismas ontologías de entrada, las salidas del matcher básico son combinadas, regularmente con alguna operación de agregación, la combinación es realizada dando valores o pesos a cada alineación de entrada, estos pesos son ajustados basados en conocimiento experto o como resultado de otra alineación.

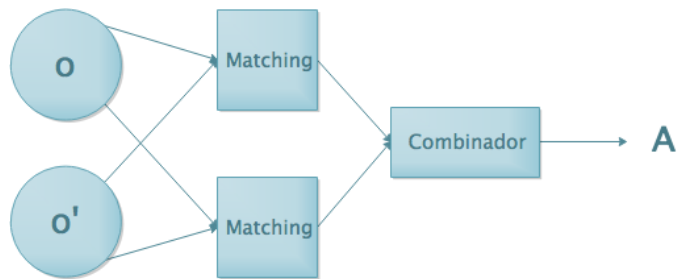


Figura 5. Estructura de un matcher en paralelo con dos técnicas básicas

Una variación o adaptación de esta estrategia es usar solo una parte de la ontología de entrada y aplicarle técnicas básicas, con esto se identifican las secciones o partes que son comunes o tienen mayor similitud con la otra ontología.

ITERACIÓN

Esta estructura de matcher utiliza reiterativamente la salida de la técnica básica como elemento de entrada para repetir el proceso, esto se repite hasta un número de repeticiones predefinido o hasta alcanzar un nivel de precisión esperado, por lo general esta estrategia usa mecanismos semiautomáticos de ejecución.

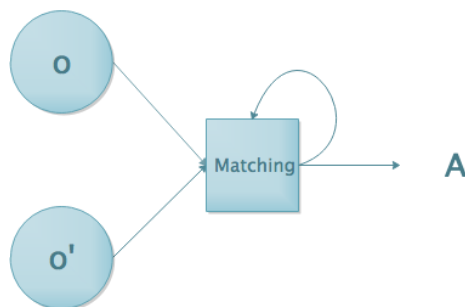


Figura 6. Estructura de un matcher iterativo

POR COMPOSICIÓN

Las estrategias descritas anteriormente se pueden combinar de múltiples formas, según el diseño del operador y puede tener componentes automáticos o semiautomáticos con intervención de un experto, a esta estructura la denominaremos estrategia por composición, donde la distribución de técnicas depende del diseño propuesto.

3.5 EL MODELO CONCEPTUAL

La siguiente definición del modelo conceptual fue tomado de (Chavarro, 2012) en el cual se hace una descripción del modelo conceptual y del modelo tecnológico el cual hace posible esta propuesta.

La creación de un matcher para *matching* ontológico en el modelo conceptual, esta soportado por la propuesta de creación de un modelo conceptual para el cambio ontológico, este modelo es una propuesta de modelado conceptual de la gestión del cambio ontológico, soportada, en un modelo teórico que extiende el metamodelo de definición de ontologías (ODM), con un metamodelo de operaciones

de cambio, y un modelo tecnológico para el desarrollo de un marco de referencia.

El metamodelo de operaciones de cambio, generaliza las operaciones que pueden ser aplicadas a una ontología y establece las correspondencias entre las operaciones y los metamodelos que representan los lenguajes de implementación de ontologías.

El marco de referencia, establece los mecanismos tecnológicos para trasladar una ontología de su lenguaje de implementación a un nivel conceptual, hacer operaciones en este nivel de abstracción y posteriormente exportarla al lenguaje de implementación seleccionado. Estas operaciones son independientes del lenguaje utilizado para la especificación de la ontología, en el caso de ontologías para el contexto de la Web Semántica, RDF y OWL. *(Chavarro & Millán, 2010)*

El modelo está compuesto por, un componente estructural para representar los elementos que constituyen una ontología, y un componente dinámico, conformado por operadores que permiten representar el cambio efectuado sobre una o varias ontologías, y por las restricciones que garantizan la consistencia del modelo. El modelo propuesto, es

la base para la construcción de un lenguaje declarativo para describir el cambio ontológico en términos de los requerimientos de cambio de la ontología.

3.5.1.COMPONENTE ESTRUCTURAL DEL MODELO CONCEPTUAL

Diferentes situaciones de modelado del mundo real, se pueden describir utilizando nodos y arcos. Tanto los nodos como las relaciones entre estos nodos se utilizan para representar jerarquías, asociaciones, o funciones, siendo común que los diversos tipos de relación coexistan en el mismo modelo. La necesidad de poder categorizar los tipos de relaciones y los tipos de nodos, motivó la introducción de una nueva estructura denominada Grafárbol.

Definición de Grafárbol: Dados los siguientes conjuntos:

- N: un conjunto de nodos tipados.
- A: un conjunto de arcos tipados cuyo tipo puede pertenecer a ArcosFuncionales o ArcosJerárquicos.
- ArcosJerárquicos: Un subconjunto de A, donde cada elemento es del tipo «es un»
- ArcosFuncionales: un subconjunto de A, donde cada elemento pertenece al conjunto «TipoDeArco».

Un Grafárbol se define, como la 3 – tupla:

$$Grafarbol = (N, A, \Psi)$$

Donde

Ψ : Si $a_k \in A$ y $n_i, n_j \in N$, entonces, $\Psi(a_k) = (n_i, n_j)$ donde (n_i, n_j) es un par ordenado, n_i nodo inicio y n_j nodo final.

Sujeto a:

1. Sea $x \in N, \exists$ un único nodopadre. $x = \text{null}$. x , es decir, el nodo raíz del árbol tiene como nodo padre a null.
2. Sean x, y, z elementos de N :
3. Si $x.y \in \text{ArcosJerarquicos} \wedge y.z \in \text{ArcosJerarquicos}$ entonces $z.x \notin \text{ArcosJerarquicos}$, es decir, el Grafárbol es a cíclico en sus relaciones jerárquicas.

Se debe anotar que la notación $x.y$ indica que x y y son nodos de un árbol y x es el padre de y .

3.5.2.REPRESENTACIÓN ESTRUCTURAL DE LA ONTOLOGÍA MEDIANTE UN GRAFÁRBOL

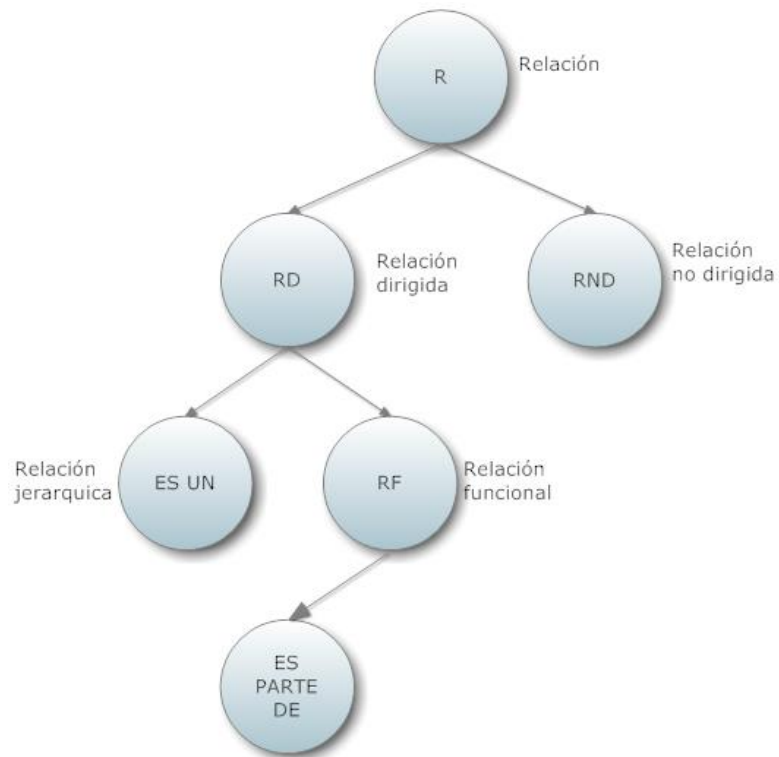


Figura 7. Jerarquía de relaciones (tomada de (Chavarro, 2012))

Una ontología, con sus operaciones y restricciones puede ser representada mediante una estructura de Grafárbol ontológico. Las relaciones utilizadas en una ontología se pueden describir mediante una jerarquía de relaciones como se muestra en la **Figura 7**, en el Grafárbol ontológico, se representan relaciones dirigidas las cuales pueden ser jerárquicas «es un» o funcionales. En el conjunto de las relaciones funcionales se destacan las relaciones «es parte de».

- Los conceptos y los tipos de datos, se representan usando `NodosConcepto C`.
- Los Individuos y los literales se representan mediante `NodosIndividuo I`,
- Las relaciones entre conceptos, entre conceptos e individuos, entre individuos, entre conceptos y tipos de datos, o entre individuos y tipos de datos o valores, se representan usando `Arcos A`.

Cuando se diferencian los tipos de nodos, se puede conformar dos niveles en el Grafárbol:

El nivel de términos, o nivel terminológico, y el nivel de instancias o individuos.

El **nivel terminológico** proporciona una estructura de árbol conformado por la relación jerárquica **«Es Un»**. Las relaciones funcionales, describen restricciones del dominio modelado.

El **nivel de instancias**, conforma un grafo a partir de las relaciones que se pueden describir entre los individuos o entre los individuos y los valores. Este nivel está interrelacionado con el nivel de términos por relaciones jerárquicas.

Cada tipo de arco, agrupa las relaciones con base en el tipo de relación que representa el mundo real; un tipo de arco refleja una vista del Grafárbol ontológico.

En la **Figura 8** se describen los arcos

- Jerarquía, arcos «es un»
- Composición, arcos «es parte de»
- Arcos funcionales «asociación»

Además en esta imagen tridimensional se ven los dos tipos de nodos (conceptos e individuos).

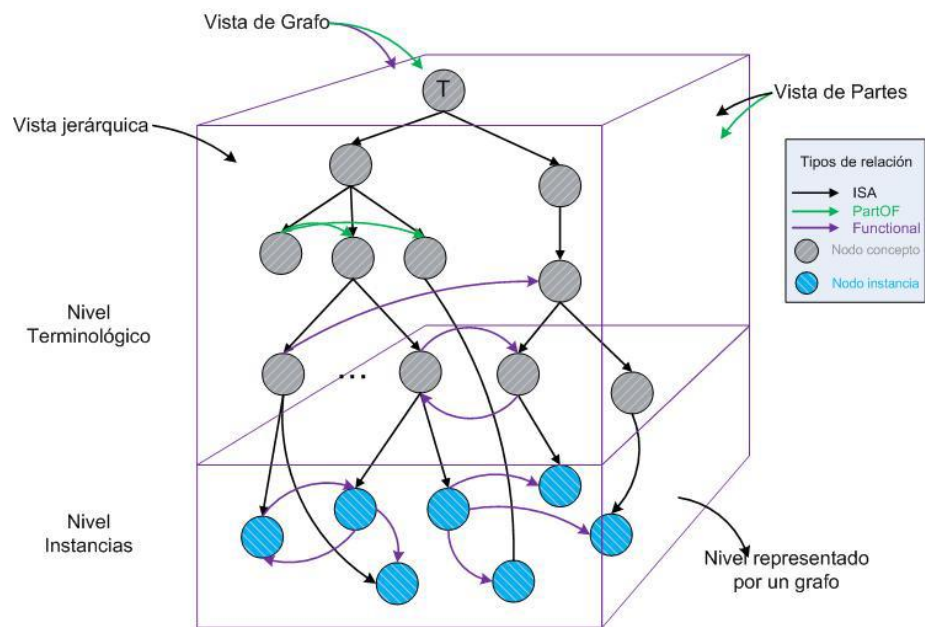


Figura 8. Grafárbol ontológico y vistas posibles (tomado de (Chavarro, 2012))

GRAFÁRBOL ONTOLÓGICO (GAO)

Un Grafárbol ontológicos se caracteriza por:

- Expresividad: En un (GAO) se pueden representar 2 tipos de nodo (Términos e individuos), dos tipos de relaciones (jerárquicas y funcionales) y en forma implícita las restricciones del modelo.
- Simplicidad: Facilidad en la compresión del modelo al presentar los conceptos de manera intuitiva.
- Minimalidad: Cada elemento del (GAO) tiene un significado diferente. No es producido por la combinación de dos o más.
- Los conceptos del modelo tiene una interpretación precisa y bien formada.

Definición de un Grafárbol ontológico:

Un Grafárbol ontológico es una 6 – *tupla*

$$G_{gfo} = (N(G_{gfo}), A(G_{gfo}), \Psi_{gfo}, E(G_{gfo}), F_{gfo}, H_{gfo})$$

Donde:

$$N(G_{gfo}) = \{n_i | n_i \in \text{TiposDeNodo}, i = 1, 2, 3, \dots, n\}, N(G_{gfo}) \neq \emptyset.$$

Es decir, $N(G_{gfo})$ es el conjunto de nodos.

$$A(G_{gfo}) = \{a_j | a_j \in TiposDeArco, j = 1, 2, 3, \dots, m\},$$

$$\text{Con } N(G_{gfo}) \cap A(G_{gfo}) \neq \emptyset$$

Disjuntos, Es decir $A(G_{gfo})$ es el conjunto de arcos.

$$\Psi_{gfo}: \text{si } a_k \in A(G_{gfo}) \text{ y } (n_i, n_j) \in N(G_{gfo}),$$

Entonces $\Psi(a_k) = (n_i, n_j)$; donde (n_i, n_j) , es un par ordenado, n_i nodo inicial y n_j nodo final.

$E(G_{gfo}) = \{e_j | e_j \text{ son etiquetas}, j = 1, 2, \dots, s\}$. Las etiquetas o anotaciones (e_0, e_1, \dots, e_s) corresponden al conjunto finito y disjunto en un lenguaje L , de nombres de etiquetas y de vértices $E(G_{gfo}) \neq \emptyset$

$F_{gfo}(n_i) = e_j$: cada par (n_i, e_j) Es único y está determinado por una función de asociación entre el elemento i del conjunto de nodos y el elemento j del conjunto de etiquetas.

$H_{gfo}(n_i) = e_n$: establece una relación funcional unica (a_m, e_n) , entre el elemento m del conjunto de arcos y el elemento n del conjunto de etiquetas.

Para representar una ontología en un Grafárbol ontológico se debe tener en cuenta:

- Los nodos criminológicos corresponden a conceptos o tipos de datos.

- Los dos individuos representen individuos nombrados o anónimos.
- Los arcos representan que las jerárquicas o funcionales entrenados de cualquier tipo.

Definición de un Grafárbol origen:

Un Grafárbol origen G_o es aquel cuyo único elemento es el nodo que representa el concepto T (thing).

$$G_{0_{gfo}} = (N(G_{gfo}), A(G_{gfo}), \Psi_{gfo})$$

Donde:

$$\begin{aligned} N(G_{gfo}) &= T \\ A(G_{gfo}) &= \{ \} \\ \Psi_{gfo} &= \{ \} \end{aligned}$$

Por simplicidad, se omiten las etiquetas y sus funciones.

3.5.3.LOS OPERADORES DE CAMBIO ONTOLÓGICO

El cambio se representa de manera independiente del lenguaje que se utiliza para implementar la ontología, mediante operadores declarativos. Un operador declarativo,

en este contexto, es aquel que permite describir el requerimiento de cambio sin el detalle de cómo se debe implementar la operación que representa.

A continuación, se presenta, en forma detallada, el conjunto de operadores, sus características, principios de diseño, clasificación, restricciones, y reglas de integridad.

Características de los operadores

Las siguientes son las características de un operador:

1. Son homogéneos en su naturaleza declarativa. Es decir, no existen operadores cuya definición sea dependiente de la implementación en el lenguaje.
2. Constituyen el núcleo de un lenguaje que permite definir las operaciones de cambio que corresponden a los niveles semántico y pragmático de una ontología, omitiendo o reduciendo los problemas asociados a las capas de representación sintáctica y léxica.
3. La expresividad del conjunto de operadores está limitada por la capacidad expresiva de la familia de operadores de la lógica de descripción que implementa, en el momento de trasladar el modelo conceptual al lenguaje de implementación.

4. Cada operador del lenguaje se puede representar en el Grafárbol.
5. Están orientados a facilitar el trabajo del usuario en el proceso de construcción y cambio de las ontologías, facilitar la actualización de la ontología y a incorporar estándares para el tratamiento de construcción y cambio ontológico.

La semejanza en la notación con el lenguaje SQL, utilizado en sistemas de bases de datos, tiene la intencionalidad de facilitar su aprendizaje por parte de programadores.

Un operador tiene la siguiente estructura sintáctica:

Operador nombre_variable {modificador [parámetro] :::};

Donde:

- El operador inicia el comando de operación.
- El nombre_variable, es el primer parámetro que recibe el operador, e indica la entidad que participa en la operación.

Una lista repetible de parámetros conformada por:

- Modificador, palabra reservada que indica la restricción que se ha de establecer sobre el operador.

- Un parámetro, que puede ser opcional, para completar la definición del modificador.
- Un carácter de terminación del operador que corresponde al punto y coma (;).

3.5.4.ESPECIFICACIÓN ESTRUCTURAL DEL LENGUAJE DECLARATIVO

En esta sección se presenta la especificación de los componentes estructurales del lenguaje, la cual incluye una descripción de las diferentes entidades que conforman una ontología, la notación utilizada en la gramática del lenguaje y algunos ejemplos, esta sección fue tomada de (Chavarro, 2012).

En el lenguaje declarativo propuesto para el modelado conceptual, las entidades son los bloques fundamentales para la construcción de las ontologías: conceptos, tipos de datos, individuos, relaciones y axiomas.

Adicionalmente, en el proceso de construcción o cambio de la ontología, surgen expresiones complejas. Estas expresiones se construyen a partir de expresiones intermedias utilizadas para delimitar rangos de datos, denotar aspectos de una relación, o delimitar conceptos. Estas

también son analizadas como parte de la especificación estructural del lenguaje y corresponden a lo que hemos denominado expresiones de rango, expresiones de relación y expresiones de concepto.

Las siguientes convenciones se utilizan para denotar diferentes partes sintácticas de ontologías en el lenguaje declarativo propuesto:

- C denota un concepto;
- EC denota una expresión de concepto;
- R denota una relación;
- ER denota una expresión de relación;
- TD denota un tipo de datos;
- ERD denota una expresión de rango de datos;
- RP denota una restricción sobre una propiedad de tipo de dato.
- I Conjunto de individuos de la ontología, a denota un individuo (con nombre o anónimo);
- It denota un literal.

CONCEPTOS (C)

En general, los conceptos pueden ser entendidos como conjuntos de individuos. Una forma de definir un concepto es asignar su nombre. Adicionalmente, un concepto también puede definirse mediante un conjunto de condiciones necesarias y suficientes, que permiten clasificar los individuos que el concepto que agrupa.

Notación gramatical:

```
ConceptName ::= ID
CONCEPTEXPR ::= CONCEPT ConceptName
SUBCLASSEXP
EQUIVALENTEXPR
DISJOINTEXPR
DISJOINTUNIONEXPR
```

Ejemplos de definiciones de conceptos, son:

Definición del concepto «DomainConcept»

```
ADD CONCEPT DomainConcept
```

En esta definición: «DomainConcept», define un concepto como subconcepto de «Thing», y corresponde a la definición de concepto atómico (A), en lógica de descripción.

Definición del concepto «País»

```
ADD CONCEPT País
SUBCLASSOF      DomainConcept
```

```
EQUIVALENTS DomainConcept
AND (America, France, England, Germany, Italy)
```

En la definición de «País» se observa el uso de la intersección entre «DomainConcept» y la lista de individuos que se describe.

Las expresiones `Disjoint` y `DisjointUnion`, son utilizadas cuando participan otros conceptos. El siguiente ejemplo ilustra su utilización.

```
ADD CONCEPT PizzaTopping
SUBCLASSOF Food
COMPOSITE DISJOINT PizzaBase, Pizza, IceCream;
```

TIPOS DE DATOS (TD)

Un tipo de dato es entendido como un espacio de valores, un espacio léxico y un conjunto de propiedades que lo restringen. Es decir, una entidad tipo de dato es un conjunto de valores, en lugar de valores individuales. Se parte del conjunto de tipos de datos definido en (*Biron, Malhotra, & Consortium, 2004*), y se adaptan las reglas de construcción y modificación.

Notación gramatical:

```
DatatypeName ::= ID
DATATYPEEXPR ::=
```

```
DATATYPE DatatypeName DATATYPEEXPR
DATATYPEPEREST
```

```
DATATYPEEXPR ::=
AS datatype |
RANGE ExprRange |
ENUM LPAREN ListID RPAREN
```

Ejemplos de la definición de un tipo de dato son:

```
ADD DATATYPE String;
ADD DATATYPE Salary AS float MIN 1;
```

En el primer ejemplo se define un tipo de dato «*String*», en el segundo ejemplo se define un tipo de dato salario, del tipo primitivo «*float*», con un valor mínimo de 1.

EXPRESIONES DE RANGO DE DATOS (ERD)

Un conjunto de valores (espacio de valores), un conjunto de formas léxicas y un conjunto de restricciones, constituyen una expresión de rango de datos. Una expresión de rango de datos (ERD), está construida a partir de un tipo de datos, el cual es en sí mismo una expresión de rango

de datos en su forma más simple, un modificador aplicado sobre un tipo de dato y uno o varios literales.

Notación gramatical:

```
ExprRange ::=
NOT ExprRange |
ExprRange MAX NUMBER |
ExprRange MIN NUMBER |
ExprRange EXACT NUMBER |
ExprRange AND ExprRange |
ExprRange OR ExprRange |
ExprRange EQUAL ExprRange |
ID
```

Los modificadores se pueden combinar para definir una restricción sobre un tipo de dato. El siguiente ejemplo incluye la combinación de varios modificadores del rango de datos.

```
ADD DATATYPE Magnitud
AS decimal
MIN 0,0001
MAX 9999,9999;
```

INDIVIDUOS (I), NOMBRADOS (a) Y ANÓNIMOS

La entidad individuos (I) de una ontología, representa el conjunto de objetos del dominio que se modela. Se utiliza la forma (a) para hacer referencia a un individuo específico

del dominio. Los individuos pueden ser nombrados, caso en el cual se utiliza un nombre para hacer referencia a un objeto específico en cualquier ontología, o anónimos cuando solo pueden ser referenciados dentro de la ontología que los contiene.

Notación gramatical:

```
IndividualName ::= ID
```

En el siguiente ejemplo, se especifica el individuo nombrado «Italy», para referenciar al país del mismo nombre y para afirmar que es miembro del concepto «País».

```
ADD INDIVIDUAL Italy
IS Country
```

EXPRESIONES DE CONCEPTOS (EC)

Una expresión de concepto (EC), se forma con una operación de cuantificación (universal, existencial), Cardinalidad (mínima, máxima, exacta), lógica (and, or, not), o de comparación, aplicada sobre un concepto, o mediante una lista de conceptos.

Notación gramatical:

```
ExprC ::= ID |
LPAREN ListExprC |
```



```

NOT ExprC |
ExprC MAX NUMBER |
ExprC MIN NUMBER |
ExprC EXACT NUMBER |
ExprC SOME ExprC |
ExprC ONLY ExprC |
ExprC ALL ExprC |
ExprC:i AND ExprC |
ExprC:i OR ExprC |
ExprC:i EQUAL ExprC
ListExprC ::= ExprC |
ExprC:i COMA ListExprC

```

Un ejemplo que ilustra el uso de expresiones de conceptos, es:

```

...
Pizza
and HasTopping
Some CheeseTopping

```

En este ejemplo, los puntos suspensivos iniciales denotan que la expresión está contenida en una cláusula del lenguaje.

RELACIONES (R)

Las relaciones expresan interacción entre conceptos, pares de individuos o individuos con literales. Las relaciones que definen un concepto como subclase de otro concepto (*Subclassof*), o que clasifican un individuo en un concepto (*is*),

son relaciones de jerarquía. Estas relaciones, en el lenguaje declarativo fueron agrupadas como modificadores a los operadores de definición de conceptos y de individuos.

Las demás relaciones que se definen son clasificadas como funcionales, se incluyen las relaciones definidas por el usuario y las subrelaciones, y se dispone de un conjunto de operadores que permite definir las características de la relación, tales como tipo de relación (simétrica, transitiva, ...), dominio, rango y relación inversa. De igual forma se puede definir una relación como disyunción, equivalencia, o unión disjunta de otra relación o grupos de conceptos.

Notación gramatical:

```

RelationName ::= ID
RELATIONEXPR ::= RELATION
RelationName SUBRELATIONEXPRR
EQUIVALENTEXPRC
DISJOINTEXPRC
DISJOINTUNIONEXPRC
TYPEEXPRR
INVERSEOFEXPRR
DOMAINEXPRR
RANGEEXPRR
SUBRELATIONEXPRR ::= SUBRELATIONOF ListExprC
TYPEEXPRR ::= TYPE TypeName:
TypeName ::= ID
DOMAINEXPRR ::= DOMAIN ExprC
RANGEEXPRR ::= RANGE ExprRange
INVERSEOFEXPRR ::= INVERSEOF RelationName

```

En el siguiente ejemplo se define una relación con sus propiedades.

```
ADD RELATION hasIngredient
TYPE Transitive
INVERSEOF isIngredientOf
DOMAIN Food
RANGE Food;
```

EXPRESIONES DE RELACIÓN (ER)

Una expresión de relación (*ER*), se forma con una operación de cuantificación (universal, existencial), Cardinalidad (mínima, máxima, exacta), lógica (and, o, not), o de comparación, aplicada sobre una relación, o mediante una lista de relaciones.

Notación gramatical:

```
ExprR ::= ID |
LPAREN ListExprC RPAREN |
NOT ExprC |
ExprC MAX NUMBER |
ExprC MIN NUMBER |
ExprC EXACT NUMBER |
ExprC SOME ExprC |
ExprC ONLY ExprC |
ExprC ALL ExprC |
ExprC AND ExprC |
ExprC OR ExprC |
ExprC EQUAL ExprC
```

Un ejemplo de una expresión de relación,

```

... hasBase
SOME PizzaBase, hasTopping
SOME FourCheesesTopping,
hasTopping
SOME TomatoTopping,
hasTopping
ONLY (FourCheesesTopping
OR TomatoTopping)

```

LITERALES (lt)

Los literales representan valores individuales de datos, por ejemplo, caracteres o enteros. En todo caso, un literal consiste de una forma léxica y un tipo de dato asociado. Esto quiere decir, que el literal «1» asociado al tipo de dato *Integer*, es diferente del literal «1» asociado al tipo de dato *float*.

Notación gramatical:

```
Literal ::= QUOTE ID QUOTE
```

Los siguientes ejemplos ilustran el uso de literales. En el siguiente ejemplo se asigna la relación «*has_salary*» entre el literal 60000 y «*salary*», el cual tiene asociado un tipo de dato «*float*» con una restricción de propiedad (Min 1).

```
ADD DATATYPE Salary
```

```
AS float
MIN 1;

ADD ASSERTION Mr_Science
has_salary 60000;
```

4

4. DISEÑO DEL OPERADOR DE MATCHING

En este capítulo se muestra el diseño del matcher construido en el modelo conceptual, se explican las técnicas que se eligieron y los procesos que se realizan para llevar a cabo el algoritmo.

Con este nuevo enfoque, el modelado conceptual, se pretende eliminar parte de los problemas al momento de buscar la interoperabilidad de ontologías. Este enfoque parte del siguiente conjunto de hipótesis, que son estudiadas en la referenciada Tesis Doctoral de Chavarro. (Chavarro, 2012)

1. Cualquier ontología puede ser representada de manera conceptual, independiente de su lenguaje de implementación y del modelo de representación del conocimiento subyacente.
2. Las operaciones de cambio ontológico pueden ser representadas de manera independiente de su lenguaje de implementación y de su modelo de representación del conocimiento.
3. Después de aplicar las operaciones de cambio, en el nivel conceptual, una nueva ontología emerge, su implementación está restringida a la expresividad del lenguaje que será utilizado.
4. La implementación de un modelado conceptual de ontologías, provee un mecanismo de administración del ciclo de vida ontológico.
5. Un sistema gestor de ontologías está soportado en meta-ontologías que modelan su ciclo de vida de manera independiente del modelo de representación del conocimiento y de su lenguaje de implementación.

Para realizar el diseño del matcher ontológico en el modelo conceptual, se realizó un análisis de las técnicas

existentes usadas en los matcher actuales clasificadas en la [sección 3.4.2](#) y explicadas en detalle en la [sección 6.2](#)

4.1. DISEÑO DE LA ARQUITECTURA DEL MATCHER

Luego de la revisión de diferentes sistemas de matching ontológico (ver [sección 6.1](#)) donde se compararon entre otras: técnicas usadas, etapas o secciones, toma de decisiones, intervención del experto, enfoque del matcher.

No se evaluaron elementos como: tecnología usada, tamaño de las ontologías en las cuales opera, lenguajes ontológicos usados

Se determina que para un sistema como OntoConcept donde entre sus características principales están: Opera en el modelo conceptual, Es de uso general, No tiene enfoque definido, Su componente estructural es el Grafárbol Ontológico. El matcher debe ser modular y permitir su crecimiento, debe ser dinámico y flexible, es decir, permitir la selección de técnicas por el experto según las ontologías a alinear, debe permitir el uso de técnicas tanto a nivel léxico como estructural, esto para aprovechar su componente estructural.

Hay que aclarar que la composición dinámica usa técnicas básicas distribuidas en secuencia y en paralelo según la configuración que el experto realice o que el mismo sistema ajuste de forma automática. La composición dinámica puede clasificarse en dos tipos: composición por oportunidad y composición pre configurada.

- **Composición por Oportunidad**

Sistema donde la próxima técnica a usar depende del resultado de la anterior, acá el experto evalúa la salida y determina según su experiencia, salida y el resultado esperado la técnica siguiente a aplicar, en esta composición no está establecido en número de iteraciones a aplicar, sistemas como FALCON AO (*Jian, Hu, Cheng, & Qu, 2005*), H-MATCH aplican parcialmente esta distribución.

- **Composición Pre configurada**

Este diseño tiene la característica que el usuario determina entre las técnicas disponibles cuales usar en y que orden, es muy útil cuando se quiere que el sistema opere semi-automáticamente sin la intervención del usuario en cada iteración, sistemas como RONDO, COMA++ (*Aumueeller, Do, Massmann, & Rahm, 2005*), PROTEGE hacen uso de esta configuración.

Un ejemplo de composición dinámica puede ser:

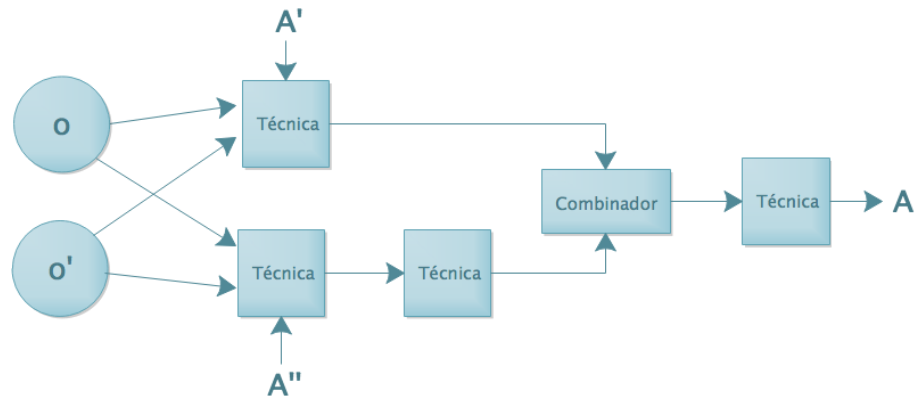


Figura 9. Ejemplo de composición dinámica

4.2. ESPECIFICACIÓN ESTRUCTURAL DEL MATCHER

Como se mencionó en la [sección 3.5.3](#) el cambio se representa de manera independiente del lenguaje que se utiliza para implementar la ontología, mediante operadores declarativos. Un operador declarativo, en este contexto, es aquel que permite describir el requerimiento de cambio sin el detalle de cómo se debe implementar la operación que representa.

Siguiendo el modelo conceptual, un operador tiene la siguiente estructura sintáctica:

Operador nombre_variable {modificador [parámetro] :::};

Se utilizarán como ejemplo dos ontologías tomadas del repositorio de casos de prueba del sistema S-Match, específicamente la prueba 7 Catalogo de Cursos Universidades (mini)⁷

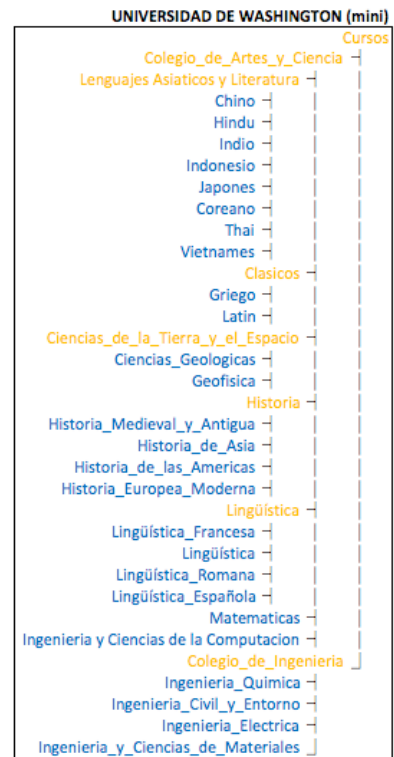
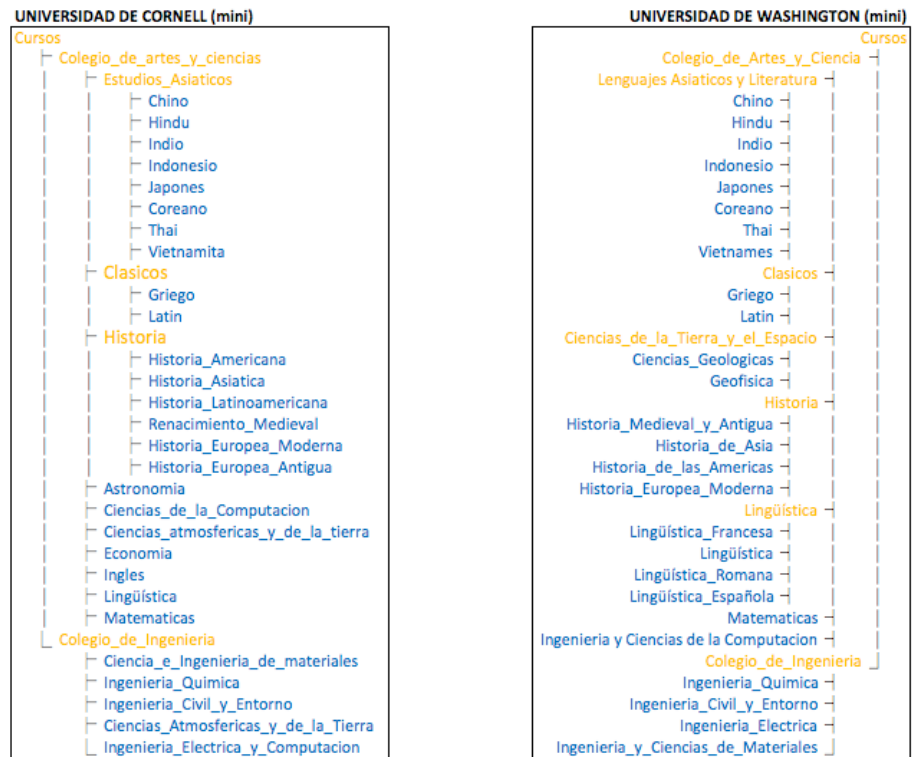


Figura 10. Ontologías de casos de prueba, universidad de Cornell y Universidad de Washington

⁷ <http://disi.unitn.it/~accord/Experimentaldesign.html>

A continuación, se describen siguiendo las normas del lenguaje declarativo los operadores que hacen parte del *matching* ontológico.

4.2.1. OPERACIONES DE PREPROCESAMIENTO (OPP)

Las operaciones de pre procesamiento son la que se ejecutan sobre una sola ontología y su producto es una ontología modificada. Corresponde a tareas previas que se deben realizar antes de iniciar el proceso de *matching*. Entre las tareas de pre procesamiento encontramos las de limpieza, normalización, supresión de prefijos, sufijos, eliminación de signos de puntuación, números y demás tareas que a nivel de elemento se deban realizar para evitar ruido o desviación en el proceso de *matching*.

Notación gramatical:

```
PREPROCESSING Técnica IPAREN Ontología [COMA ListArgumentos] DPARENT

Técnica ::= Nombre Técnica de pre procesamiento
Ontología ::= ID
ListArgumentos ::= Argumento |
Argumento COMA ListArgumentos
```

Ejemplo de función de pre procesamiento:

PREPROCESING CaseNorm (OntologyCornell)

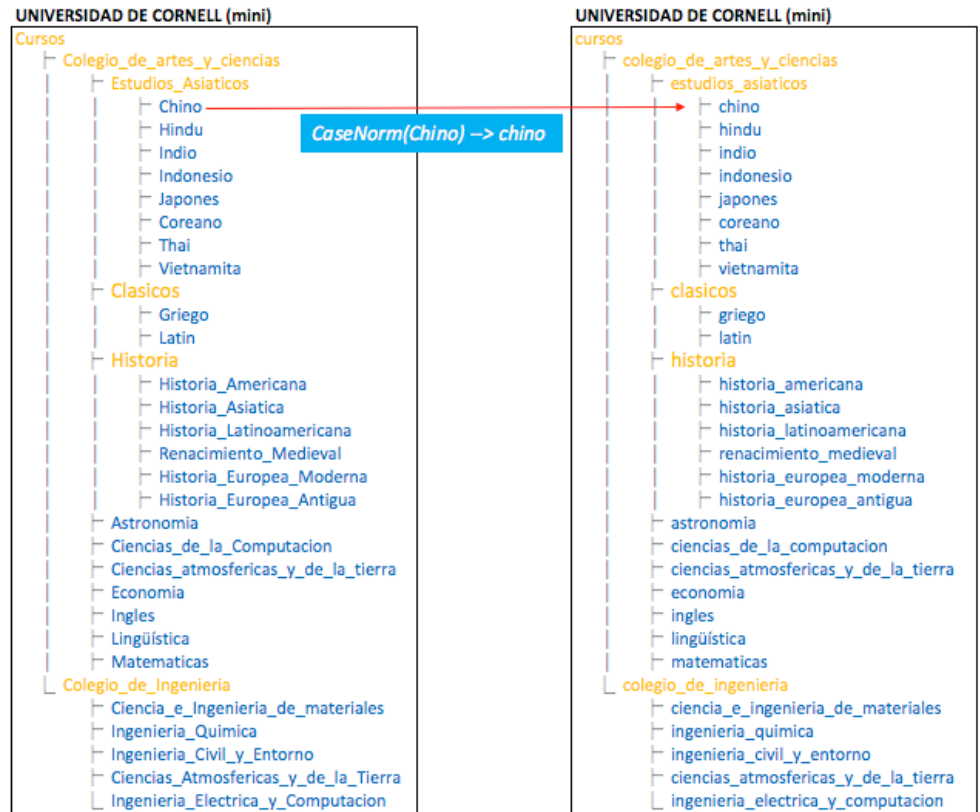


Figura 11. Aplicación de la funcion CaseNorm

Donde:

`CaseNorm` hace referencia a la operación de normalización de cadenas de texto; convierte todas las cadenas de texto a minúscula, produciendo como salida una ontología donde cada carácter de las cadenas de texto esta expresado en minúscula, no tiene parámetros adicionales, por los cuales

no tiene sino un argumento y es el nombre de la ontología a aplicar la función.

4.2.2. OPERACIONES DE PROCESAMIENTO DE *MATCHING*

El operador de emparejado ontológico (*MATCHING*), genera como salida un conjunto que corresponde a las equivalencias encontradas en los dos conjuntos. Requiere como parámetros: la selección de la técnica que se van a utilizar, las dos ontologías a aplicar la técnica y una alineación como parámetro opcional.

```
MATCHING ONTOLOGY Técnica
IPAREN Ontología1 COMA Ontología2 DPARENT [ALINEACION_ENTRADA]

Técnica ::= Nombre Técnica IPARENT [ListArg] DPARENT
ALINEACION_ENTRADA ::= ID
```

La salida del operador de `MATCHING ONTOLOGY`, es un conjunto denominado **alineación**, este a su vez puede convertirse en un parámetro de entrada a otro operador similar, haciendo de este un proceso recursivo hasta obtener una alineación consistente con el resultado esperado.

Es importante aclarar que la salida real de una operación de `MATCHING ONTOLOGY` es una matriz de similitud donde se

compararon todos los elementos de una ontología con los de la otra ontología y se debe realizar un paso adicional para convertir esta matriz en una alineación, este paso está inmerso en el proceso de generación de la alineación.

Ejemplo de matriz de similitud

	Chino	Thai	Latin	Historia_de_las_Americas	Lingüística	Matematicas	Ingenieria_Civil_y_Entorno
Chino	1,00	0,05	0,02	0,01	0,00	0,00	0,01
Thai	0,05	1,00	0,20	0,01	0,15	0,02	0,00
Latin	0,02	0,20	1,00	0,01	0,01	0,01	0,00
Historia_Americana	0,00	0,00	0,01	0,93	0,02	0,01	0,00
Astronomia	0,01	0,00	0,00	0,00	0,00	0,10	0,10
Economia	0,01	0,00	0,00	0,00	0,00	0,10	0,10
Lingüística	0,01	0,00	0,00	0,00	1,00	0,10	0,10

Figura 12. ejemplo de matriz de similitud

	Chino	Thai	Latin	Historia_de_las_Americas	Lingüística	Matematicas	Ingenieria_Civil_y_Entorno
Chino	1,00	0,05	0,02	0,01	0,00	0,00	0,01
Thai	0,05	1,00	0,20	0,01	0,15	0,02	0,00
Latin	0,02	0,20	1,00	0,01	0,01	0,01	0,00
Historia_Americana	0,00	0,00	0,01	0,93	0,02	0,01	0,00
Astronomia	0,01	0,00	0,00	0,00	0,00	0,10	0,10
Economia	0,01	0,00	0,00	0,00	0,00	0,10	0,10
Lingüística	0,01	0,00	0,00	0,00	1,00	0,10	0,10

Figura 13. Ejemplo de matriz de similitud son los valores seleccionados para la alineación

Luego de seleccionar los valores que superan un umbral pre configurado se determina cual es la alineación de salida del operador de procesamiento de *matching* ontológico.

```
...
O1(chino) → O2(chino): 1,00
O1(Thai) → O2(Thai): 1,00
O1(Latin) → O2(Latin): 1,00
O1(Historia_Americana) → O2(Historia_de_las_Americas): 0,93
O1(Lingüística) → O2(Lingüística): 1,00
...
```

Las técnicas a utilizar en esta etapa se pueden clasificar en dos tipos: las que operan a nivel léxico y las que operan a nivel de estructura. Entre las técnicas posibles están:

- **A nivel de léxico:** Distancia hamming, Semejanza n-grama, Distancia de Edición, Medida jaro-winkler, Similitud Coseno, Distancia de Camino.
- **A nivel de estructura:** semejanza Wu-Palmer, Proximidad de Vecindad, Distancias Jerárquicas, Similitud de Herencia por Descendientes y Similitud por Hermandad.

El experto puede combinar estas técnicas como se mencionó en la **sección 4.1** de forma secuencial o paralela, dependiendo de la técnica a utilizar, esta puede contener parámetros para ajustar su operación, como son: umbrales, restricciones de iteraciones, tiempo, etc.

La alineación de entrada es un parámetro opcional y se usa para dinamizar o potencializar la técnica usada, un ejemplo de esto puede ser la aplicación de una técnica solamente en los elementos que superaron un umbral en la alineación de entrada.

Ejemplo de operación de procesamiento *matching*

```
MATCHING ONTOLOGY NGRAM (3) (OntologyCornell, OntologyWashington)
```

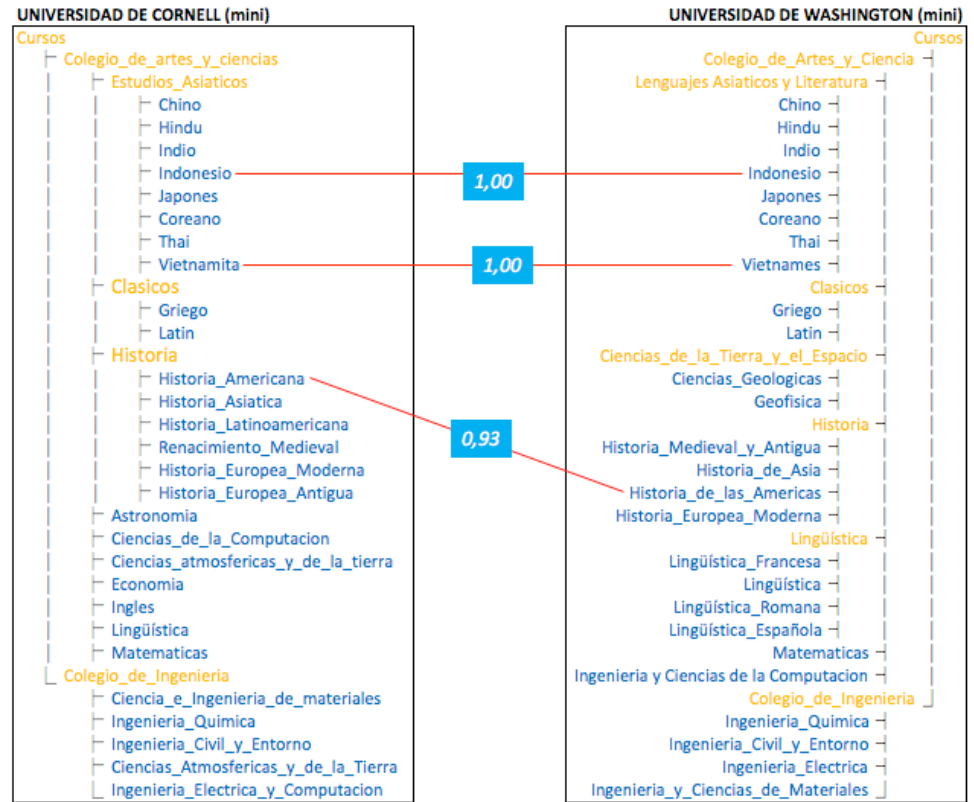


Figura 14. aplicación de la operación de procesamiento ngram

En este ejemplo se aplica la técnica básica de nivel léxico NGRAM «n-grama» con parámetro «3», indicando que se compararan en las dos ontologías de entrada «Ontology-Cornell» y «OntologyWashington» cadenas de 3 caracteres, el resultado de aplicar esta técnica es una alineación.

4.2.3. OPERACIONES DE COMBINACION

Las operaciones de combinación son las usadas cuando tenemos varias alineaciones y queremos obtener una a

partir de algún método combinatorio, estas operaciones de combinación se usan en conjunto con las operaciones de procesamiento de *matching*.

Las operaciones de combinación reciben como parámetro un conjunto de alineaciones de entrada y producen una única alineación de salida, entre las operaciones de combinación que se pueden aplicar están: suma ponderada, promedio, máximo, mínimo, mediana, normalización triangular, entre otras.

```
COMBINATION ALIGNMENT COMBINACION
IPARENT ListaAlineaciones DPARENT

COMBINACION ::=
NombreTecnica IPARENT ListArg DPARENT
ListaAlineaciones ::= Alineacion |
                    Alineacion COMA ListaAlineaciones
```

Ejemplo de aplicación de técnica de combinación:

```
COMBINATION ALIGNMENT SUMA_PONDERADA (0.2,0.8)
(Alineacion1, Alineacion2)
```

Donde:

Alineacion₁ y Alineacion₂ son alineaciones producidas por la salida de una técnica de matching y SUMA_PONDERA es una

función de agregación donde toma los pesos de las alineaciones de entrada y usa los parámetros recibidos como pesos de ponderación en una única alineación de salida.

Ontology Cornell	Ontology Washington	Alineación₁	Alineación₂	Suma Pon- derada (0.2,0.8)
Chino	Indio	0,3719	0,1868	0,2238
Hindu	Japones	0,4674	0,5659	0,5462
Coreano	Coreano	0,4977	0,7582	0,7061
Vietnamita	Thai	0,2872	0,7956	0,6940
Latin	Cien- cias_Geolo- gicas	0,0784	0,6284	0,5184
Histo- ria_Asiatica	Geofisica	0,2335	0,0149	0,0586
Renaci- miento_Me- dieval	Histo- ria_de_Asia	0,3092	0,0689	0,1170
Historia_Eu- ropea_Mo- derna	Historia_Eu- ropea_Mo- derna	0,0899	0,6324	0,5239
Astronomia	Lingüística	0,3875	0,4103	0,4057

Cien- cias_de_la_ Computacion	Ingenieria y Ciencias de la Compu- tacion	0,3090	0,7888	0,6928
Ingles	Ingenie- ria_Quimica	0,9737	0,4116	0,5240
Matematicas	Ingenie- ria_Ci- vil_y_En- torno	0,0071	0,4549	0,3654
Ingenie- ria_Quimica	Ingenie- ria_Electrica	0,8828	0,5369	0,6061
Ingenie- ria_Ci- vil_y_En- torno	Ingenie- ria_y_Cien- cias_de_Ma- teriales	0,8780	0,6359	0,6843

Tabla 11. Ejemplo de aplicación de operación de combinación

4.2.4.OPERACIONES DE FILTRO

Las operaciones de filtro son operaciones que se aplican sobre una alineación produciendo como salida una alineación modificada, utiliza algunas técnicas como fortalecimiento o debilitamiento de la alineación de entrada.

Estas operaciones de filtro son útiles para descartar valores fuera de lo normal estadísticamente hablando o para darle mayor peso a una alineación, permite maximizar la calidad de una alineación

Algunas técnicas son: función sigmoidea, función escalar.

```
FILTER ALIGNMENT FILTRO
IPARENT Alineacion DPARENT

FILTRO ::= NombreTecnica IPARENT ListArg DPARENT
```

Ejemplo de aplicación de técnica de filtro:

```
FILTER ALIGNMENT SIG () (Alineacion1, Alineacion2)
```

Ontology Cornell	Ontology Washington	Alineación	Sigmoide
Chino	Indio	0,2238	0,0594
Hindu	Japones	0,5462	0,6136
Coreano	Coreano	0,7061	0,8870
Vietnamita	Thai	0,6940	0,8743
Latín	Ciencias_Geologicas	0,5184	0,5458
Historia_Asiatica	Geofisica	0,0586	0,0120

Renacimiento_Medieval	Historia_de_Asia	0,1170	0,0212
Historia_Europea_Moderna	Historia_Europea_Moderna	0,5239	0,5594
Astronomia	Lingüística	0,4057	0,2803
Ciencias_de_la_Computacion	Ingenieria y Ciencias de la Computacion	0,6928	0,8731
Ingles	Ingenieria_Quimica	0,5240	0,5598
Matematicas	Ingenieria_Civil_y_Entorno	0,3654	0,2065
Ingenieria_Quimica	Ingenieria_Electrica	0,6061	0,7428
Ingenieria_Civil_y_Entorno	Ingenieria_y_Ciencias_de_Materiales	0,6843	0,8633

Tabla 12. Ejemplo aplicación operación de filtro

Como las operaciones de filtro producen una alineación pueden ser usadas como parámetro de ingreso a una operación de procesamiento.

Las operaciones descritas anteriormente:

- Operaciones de pre procesamiento

- Operaciones de procesamiento de *matching*
- Operaciones de combinación
- Operaciones de filtro

Son los elementos con los cuales un experto puede configurar un matcher dependiendo de la disponibilidad de técnicas, características de las ontologías, recursos disponibles como alineaciones previas, u cualquier otra variación en el entorno. Un ejemplo de matcher puede ser:

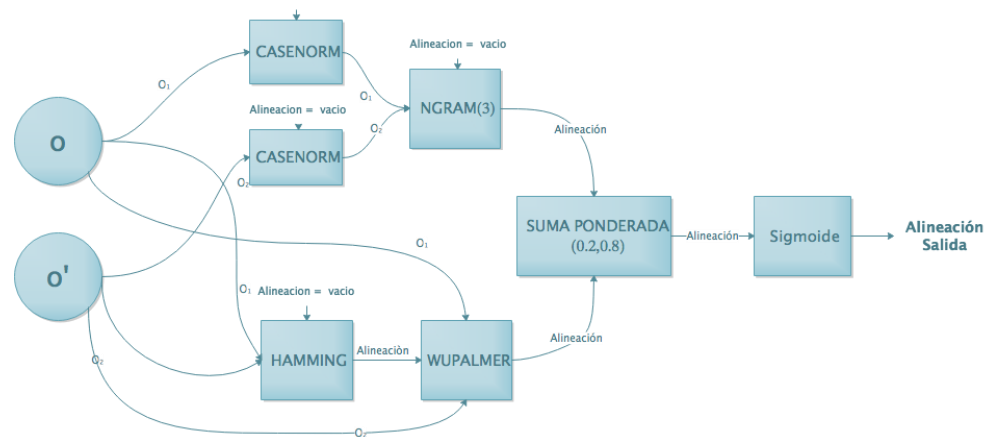


Figura 15. Ejemplo de configuracion de matcher

La notación de esta configuración es:

```

FILTER ALIGNMENT SIG (
  COMBINATION ALIGNMENT SUMAPONDERA (0.2, 0.8) (
    MATCHING ONTOLOGY NGRAM (3) (
      PREPROCESSING CASENORM (OntologyCornell),
      PREPROCESSING CASENORM (OntologyWash)
    ),
    MATCHING ONTOLOGY WUPALMER () (
      (OntologyCornell, OntologyWash)
    )
  )
)
  
```

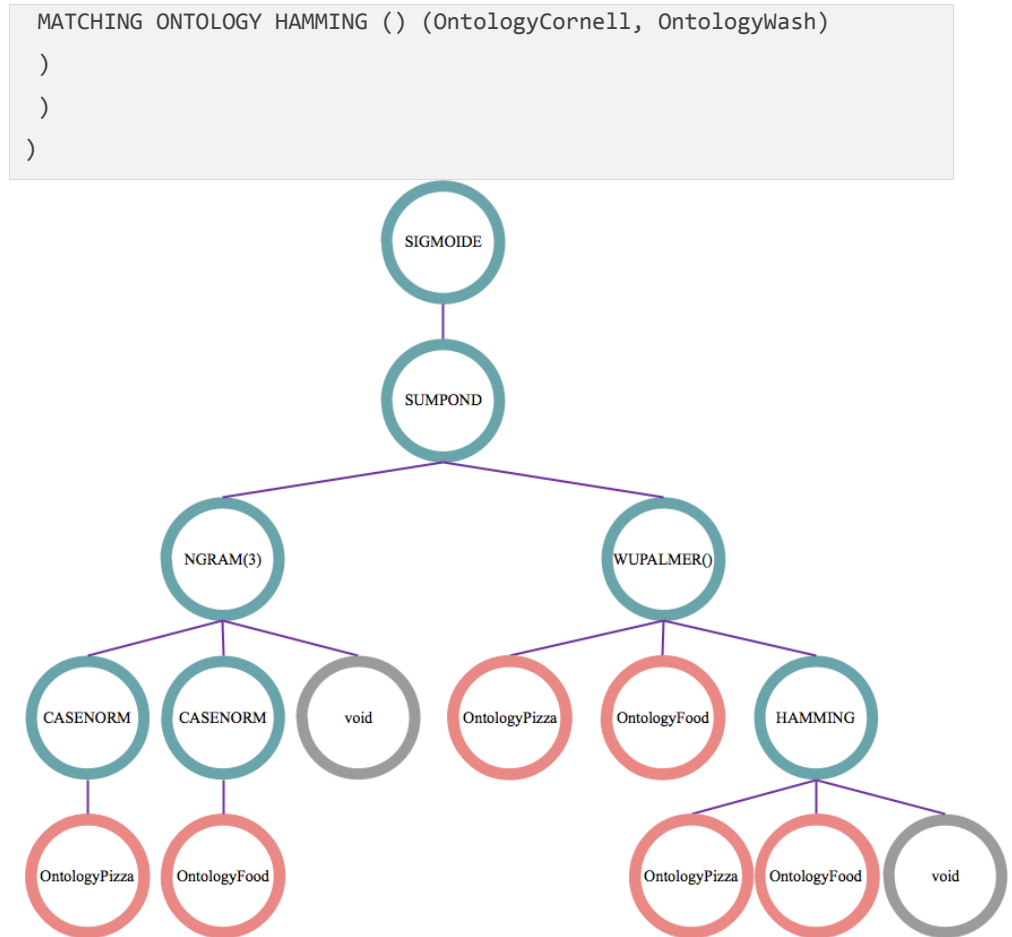



Figura 16. Árbol de operaciones del matcher de ejemplo

4.3. MODELO TECNOLÓGICO

El matcher diseñado está caracterizado por basarse en semejanza de clases, y permite la configuración dinámica aprovechando el conocimiento experto del usuario. El matcher realiza cuatro etapas:

1. Etapa de preprocesamiento
2. Etapa de procesamiento

3. Etapa de combinacion

4. Etapa de filtro

Como se explicaron en la **sección 4.2**

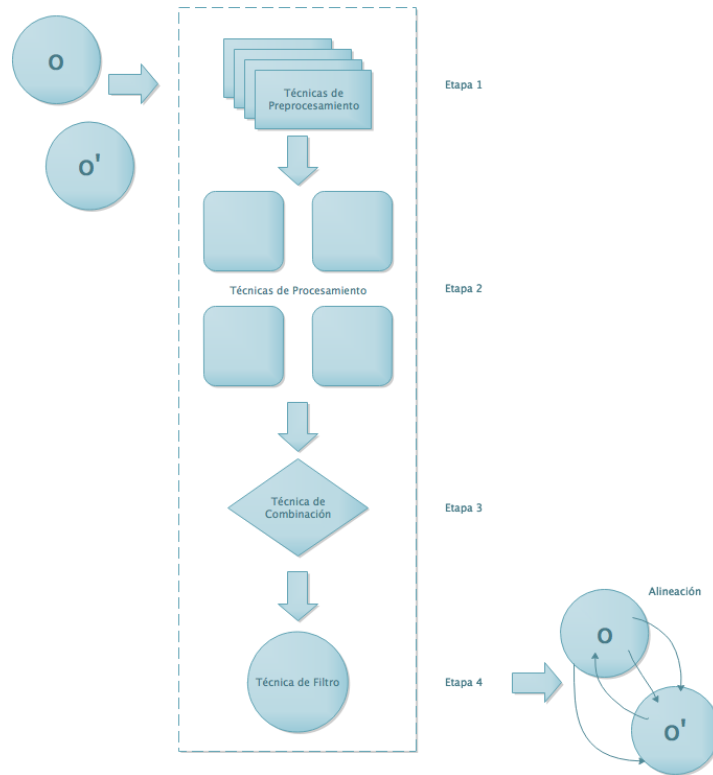


Figura 17. Etapas del matcher

Este matcher hace parte del marco de referencia para cambio ontológico, propuesto por Chavarro ([Chavarro, 2012](#)), el cual presenta una descripción modular del Marco de Referencia (framework OntoConcept) para la gestión conceptual del cambio en ontologías. Tecnológicamente, el marco de referencia es un servicio web, extensible, portable, con especificación modular basada en WSDL 2.0, y

guiado por los principios del software libre, como se ve en la **Figura 18**

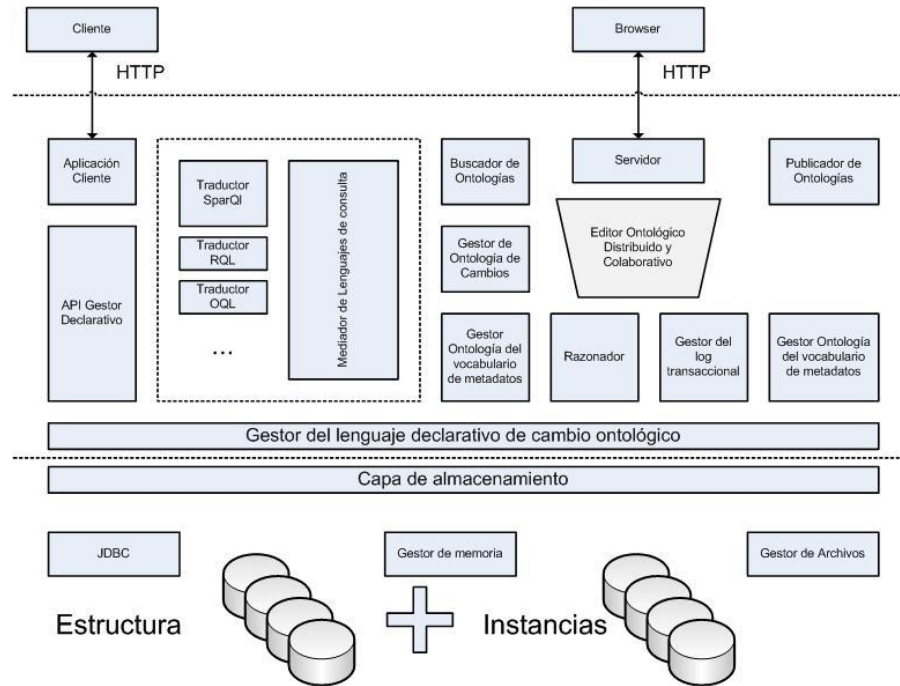


Figura 18. Arquitectura marco de referencia para la gestión conceptual del cambio ontológico. Tomado de (Chavarro, 2012)

En trabajos posteriores se definió la línea base arquitectural del framework OntoConcept (Salazar, 2015) el cual se muestra en la **Figura 19**

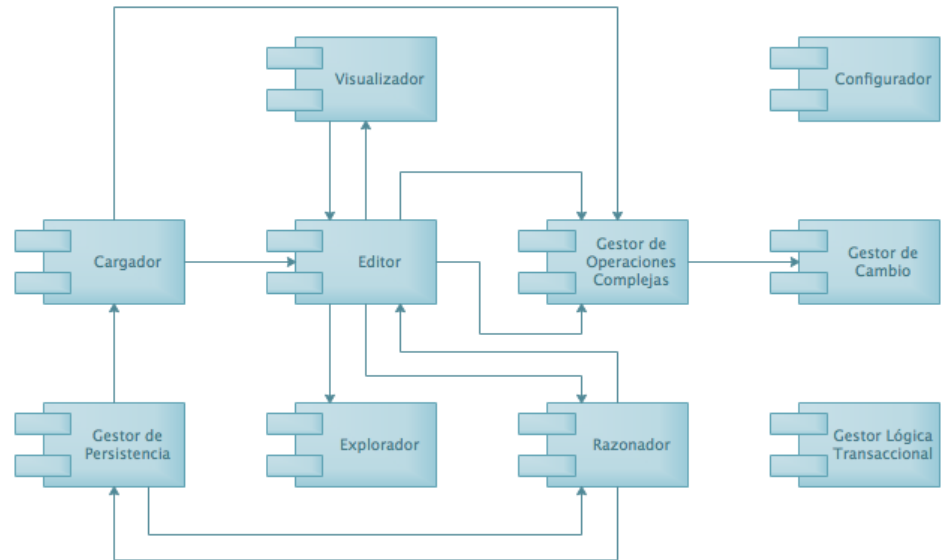


Figura 19. Diagrama de componentes, adaptado de (Salazar, 2015)

Tomando como base el diseño realizado por (Salazar, 2015) se desarrolla en el componente «Gestor de Operaciones Complejas» para continuar el desarrollo y crecimiento del framework OntoConcept.

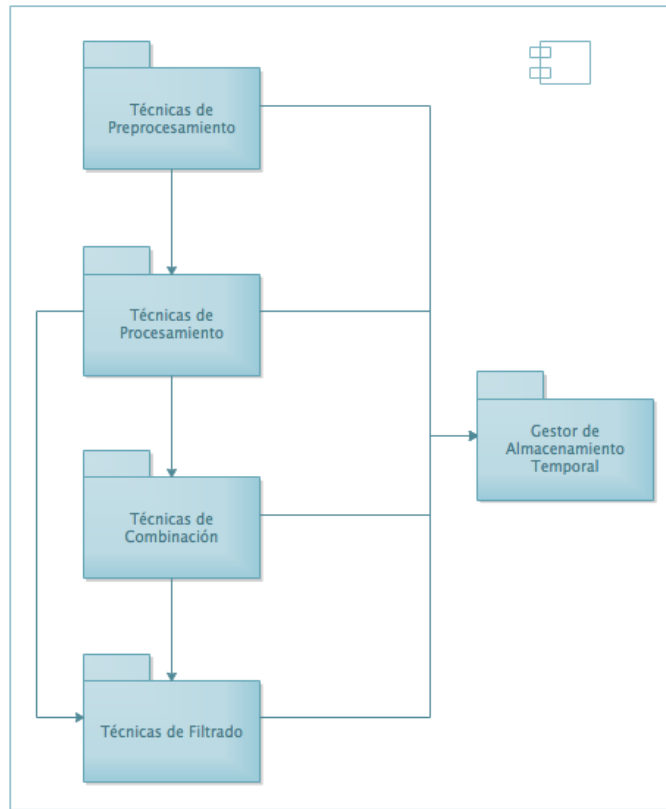


Figura 20. Vista interna del componente gestor de operaciones complejas

Como lo menciona (Chavarro, 2012), las operaciones como *matching* ontológico son consideradas operaciones complejas, porque requiere de la intervención de más de una ontología.

En cada uno de los paquetes planteados se desarrollarían las técnicas apropiadas.

Así:

Paquete técnicas de pre procesamiento

- Normalización Mayúscula/Minúscula
- Supresión de tildes
- Espacios
- Tabulaciones
- Guiones
- Dígitos
- Signos de Puntuación
- Normalización lingüística
- Tokenización
- Lematización
- Eliminación de Conectores

Paquete técnicas de procesamiento

- Distancia de Hamming
- Similitud n-gramas
- Distancia de Levenshteing
- Medida de Jaro-Winkler
- Distancia Manhattan
- Distancia de Mikowski
- Similitud de Coseno

- Disimilitud topológica estructura
- Similitud Wu-Palmer

Paquete de Técnicas de Combinaciones

- Suma ponderada
- Promedio
- Máximo
- Mínimo
- Mediana
- Normalización triangular

Paquete de Técnicas de Filtro

- Función Sigmoidea
- Función Escalar

El modelo tecnológico propuesto para implementar el matcher ontológico que operan en el modelo conceptual, sigue la línea de desarrollo planteada inicialmente por (Chavarro, 2012) y que complemento (Salazar, 2015) en su trabajo. Los puntos principales para hacer posible esta implementación

son: (1). que sea compatible y adaptable al trabajo inicialmente propuesto, (2) que sea modular y reusable, (3) que permita comunicación entre los módulos del framework OntoConcept (4) que permita el desarrollo e implementación incremental de técnicas tanto a nivel léxico como estructural. (5) Compatible con tecnologías web, para hacer un framework OntoConcept distribuido y de acceso concurrente.

5

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

5.1. CONCLUSIONES

El desarrollo del presente trabajo se basó en su mayoría en el trabajo de (Chavarro, 2012) Marco de Referencia para la Gestión del Cambio Ontológico, y se plantearon varios retos, como lo son el análisis crítico de los principales sistemas de *matching* entre los cuales se incluyen: COMA (Do & Rahm, 2002), GLUE (Doan, Madhavan, Domingos, & Halevy, 2004), S-MATCH (Giunchiglia, Shvaiko, & Yatskevich, 2004), OLA (Kengue, Euzenat, & Valtchev, 2007), FALCON AO (Hu & Qu, 2008), SAMBO (Lambrix & Tan, 2011), AROMA (David, 2009), RIMON (Li, Tang, Li, & Luo, 2009), AGREEMENT MAKER (Cruz, Antonelli, & Stroe, 2009), ASMOV (Jean-Mary, Shironoshita, & Kabuka, 2010), TAXOMAP (Hamdi, Safar, Niraula, & Reynaud, 2009), PROMPT (Noy & Musen, 2003), DSSIM (Nagy,

Vargas-Vera, Stolarski, & Motta, 2008). Con este análisis se pudo determinar que:

- Los *Matching* basados en esquemas tienen más investigación que los matcher basados en instancias.
- La mayoría de los sistemas se enfocan en algunos dominios específicos, solo unos pocos como: COMA, COMA++, Similaty flooding, S-Match son de uso general.
- La mayoría de los sistemas usan de entrada dos ontologías, solo unos pocos usan más de dos ontologías de entrada como: DCM, Wise Integrator
- La mayoría de los enfoques manejan estructuras de tipo árbol, solo unos pocos usan estructuras de tipo grafo como: COMA, CUPID, COMA++, OLA.
- La mayoría de los enfoques buscan relaciones uno a uno, solo algunos sistemas como: IMAP, DCM, buscan relaciones de uno a muchos y de muchos a muchos
- La mayoría de los sistemas buscan relaciones de confianza entre 0 y 1, solo unos se enfocan en otro tipo de relaciones como: equivalencia, subsunción, por ejemplo: CTXMATCH, S-MATCH.

Con este análisis y con los elementos expuestos en la **sección 4.1** se determinó que el *matching* empleara una composición dinámica por capas enfocada en el esquema sin discriminación de tipo de relación,

El diseño modular de OntoConcept permite el diseño de operadores por capas independiente del lenguaje de implementación

La arquitectura desarrollada en (Chavarro & Millán, 2010), permite intuir la viabilidad de la construcción del operador de *matching*, con las características planeadas independiente del lenguaje de implementación.

Las pruebas realizadas con las técnicas seleccionadas y diferentes configuraciones estructurales planteadas, son pruebas para verificar la viabilidad de la propuesta, estas pruebas comprenden un ejercicio regular con límites establecidos entre dos ontologías conocidas y con resultados conocidos, por lo tanto, no pueden ser tomadas como pruebas formales del modelo. La demostración del cumplimiento formal del modelo está por fuera del alcance del presente trabajo, queda como un ejercicio que puede ser desarrollado en un nivel de educación formal superior.

5.2. TRABAJOS FUTUROS

Con el diseño del operador de *matching* ontológico propuesto y validado de forma teórica, donde se plantean cuatro capas, el trabajo futuro a desarrollar es la creación de técnicas en cada una de ellas, como las planteadas en la [sección 4.3](#).

Adicional a las capas planteadas existe una quinta capa que nos permitiría darle al usuario experto un último punto de interacción, en esta capa el sistema expondría sugerencias y conflictos que no se pudieron resolver con las técnicas configuradas para que usuario determine basado en el conocimiento experto de las ontologías de entrada.

Con el desarrollo del *Framework OntoConcept* y la extensión a más leguajes de definición de ontologías se puede extender también el *matching* ontológico según la capacidad expresiva de estos lenguajes.

Para finalizar, el proceso de *matching* ontológico busca encontrar correspondencias entre dos ontologías, pero el trabajo no finaliza aquí, con esta correspondencia se debe determinar la operación compleja que se ejecutará sobre las ontologías, como fusión, mezcla, integración y aun se

debe continuar trabajando para que estas operaciones sean una realidad en OntoConcept.

6

6. ANEXOS

6.1. SISTEMAS DE *MATCHING*

A continuación, se expondrán una serie de propuestas que han participado en campañas propuestas por la OAEI para su evaluación. Se seleccionaron soluciones de la última década aproximadamente. Estos trabajos fueron citados en su mayoría en el libro *Ontology Matching* (Euzenat & Shvaiko, 2007) o en artículos como: (Shvaiko & Euzenat, 2005), (Granitzer, Sabol, Onn, Lukose, & Tochtermann, 2010), (Euzenat, Meilicke, Stuckenschmidt, Shvaiko, & Trojahn, 2011), (Choi, Song, & Han, 2006).

6.1.1.COMA

Combination of Match algorithms COMA (Do & Rahm, 2002) es una estrategia hibrida que combina diferentes medidas

para relacionar: esquemas XML y esquemas relacionales. El sistema permite que un usuario compare los resultados, se ha implementado un gestor de base de alineamientos para reutilizaciones.

A nivel léxico emplea diferentes medidas: q-gram, similitud por sinonimia, uso de la totalidad de cada elemento, etc.

A nivel estructural, comprueba el número de descendientes y hojas entre los diferentes elementos como valor inherente de similitud. Incorpora diferentes estrategias de combinación donde se aplican medias, medias ponderadas, selección por rango y el coeficiente de DICE.

En la primera versión el objetivo es el esquema independientemente del lenguaje. COMA++ (*Aumüller et al., 2005*) es una extensión de COMA con la compatibilidad con el estándar OWL-lite, al mismo tiempo admitiendo SQL y W3C XSD.

6.1.2.GLUE

GLUE (*Doan et al., 2004*) es una solución basada en la combinación de dos tipos de heurística para complementar los resultados. El algoritmo calcula la probabilidad conjunta de dos conceptos en sus respectivas taxonomías, mediante

la similitud Jaccard, Así, determina aquellos individuos pertenecientes a la unión de ambas clases.

Las dos técnicas heurísticas se basan en la proximidad de vecindad producida por la relajación del etiquetado y en una serie de restricciones. La relajación del etiquetado en clases vecinas se produce cuando el nombre de las subclases contiene el nombre de la clase padre. Las restricciones sobre esta heurística se basan en casos específicos como: dos clases serian similares si sus parientes son similares o al menos en algún grado, si todos las subclases de un nodo X coinciden con todas las subclases de un nodo Y entonces X y Y son similares.

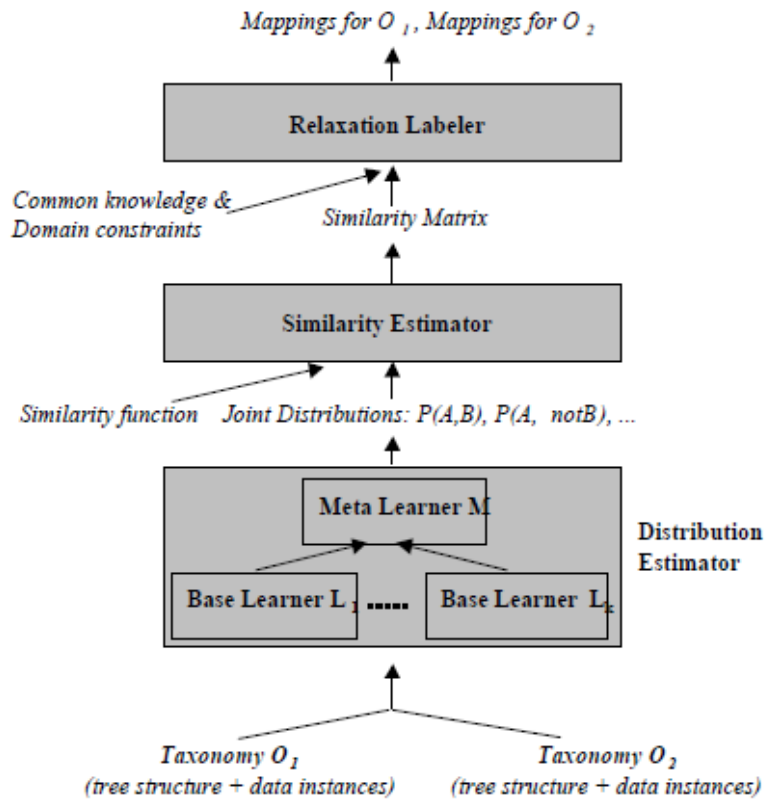


Figura 21. Arquitectura GLUE (tomado de (Doan et al., 2004))

Los autores comentan que aplican técnicas de machine learning pero no explican, ni definen el tipo de aprendizaje, ni los datos históricos utilizados para tal fin.

Los alineamientos no son representados mediante ningún lenguaje específico.

6.1.3.S-MATCH

S-Match (Giunchiglia et al., 2004), (Shvaiko, Giunchiglia, & Yatskevich, 2010) es un algoritmo caracterizado por basarse

en la similitud de clases en vez de etiquetas. Uno de los primeros en definir tipos de relaciones como resultado del alineamiento en vez de un valor numérico y en utilizar el significado de las clases para mapear.

Su solución se basa en dos ideas:

- Concepto de una etiqueta
- Concepto en un nodo.

El primero hace referencia al conjunto de instancias que se podrían englobar en esa etiqueta y la segunda idea hace referencia al conjunto de instancias que se podrían agrupar a un cierto nodo con referencia a una etiqueta y a una cierta posición dentro del grafo.

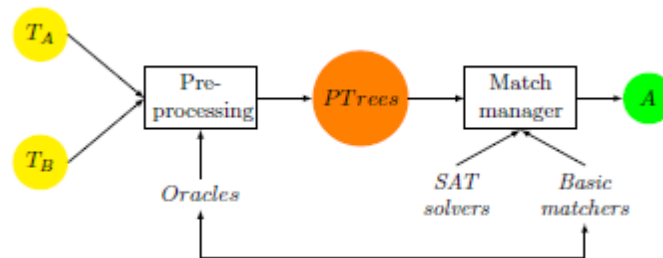


Figura 22. Arquitectura de S-Match (tomado de (Shvaiko et al., 2010))

S-Match realiza cuatro pasos:

- El primero determina que concepto se haya detrás de cada etiqueta mediante el uso de WordNet⁸. Las etiquetas compuestas se contemplan como la unión de ambas excepciones
- El segundo paso consiste en determinar la acepción en función de la posición en el árbol, desde un punto de vista jerárquico para la clasificación y subsunción de palabras.
- Tercer paso, computa la similitud en base al etiquetado de las clases aplicando una serie de comparadores en el siguiente orden: prefijos, sufijos, edit distance, q-gram, Corpus textual, WordNet, distancia jerárquica, el gloss⁹ de WordNet, la versión extendida del gloss de WordNet, comparación del gloss, comparación extendida del gloss, comparación semántica del gloss y comparación semántica extendida del gloss. Los cinco primeros son basados en comparación de caracteres. Los dos siguientes en

⁸ <https://wordnet.princeton.edu/>

⁹ Definición y/o sentencias de ejemplo

comparación de la acepción, y el resto son basados en el gloss.

- El cuarto y último paso consiste en determinar las relaciones entre los conceptos de nodos. La correcta elección de relaciones entre conceptos la han basado en un problema de Satisfacibilidad booleana (SAT) (Sánchez & Béjar, 2000). El tipo de alineamiento semántico se convierte en una tabla de verdad con el objetivo de encontrar el vacío o una contradicción. Además, para simplificar la magnitud de relaciones han incorporado varias técnicas de simplificación de proposiciones.

Los alineamientos se definen en base a cuatro tipos de relaciones: equivalencia ($=$), subsunción (más general y menos general), sin correspondencia (\perp) y con solapamiento (\cap). Este último tipo no lo han implementado ya que WordNet no provee tal categoría. Los alineamientos son representados mediante XML.

6.1.4.OLA

OWL Lite Alignment OLA (Kengue et al., 2007),(Soriyan, Gambo, & Ikono, 2012), es una herramienta de código abierto que

implementa el algoritmo OLA, el objetivo principal detrás de esta herramienta es el de mejorar las alineaciones ontológicas expresadas en OWL, con énfasis en OWL-LITE, OLA se basa en técnicas de mapeado de esquemas e instancias.

Este tipo de representaciones poseen un mapeado con un tipo especial de grafos llamado OL-Graph donde los vértices corresponden a las entidades y los arcos representan las relaciones entre entidades. El conjunto de diferentes vértices representan las clases, relaciones, propiedades, propiedades de las instancias, y tipos de datos, valores de datos y restricciones.

El problema se reduce a encontrar similitudes entre nodos de ambos grafos. La similitud resultante, un valor entre 0 y 1, es la suma ponderada de las diferentes similitudes consideradas. Estas son propiedades datatype y object, la firma de la propiedad, tipo, el valor de los literales y la etiqueta.

La propuesta tal como comentan los autores se centra en la descripción estructural de los elementos, es decir, mismas descripciones hacen similares a esos conceptos. La

correcta justificación del cálculo de las diferentes similitudes esta vagamente explicado. No existe una evaluación o comparación con otras iniciativas.

6.1.5.FALCON AO

Falcon-AO (*Jian et al., 2005*) es una composición de dos algoritmos de mapeado: uno léxico (LMO) y el otro estructural (GMO). El algoritmo lingüístico combina a su vez otras dos medidas: la distancia de edición de la etiqueta y una distancia heurística basada en la aparición de términos acontecidos en los comentarios y en las diferentes etiquetas que pueda presentar las clases vecinas de la clase. Ambas medidas son unificadas mediante una suma ponderada por pesos experimentales. La medida estructural se basa en un cálculo propio de anteriores trabajos en comparar grafos. La medida se basa en comparar tripletas con información estructural similar: sujeto, predicado y objeto. Según los autores, para que el proceso de similitud estructural sea más preciso se utiliza la medida de similitud léxica mediante la fórmula de similitud de coseno.

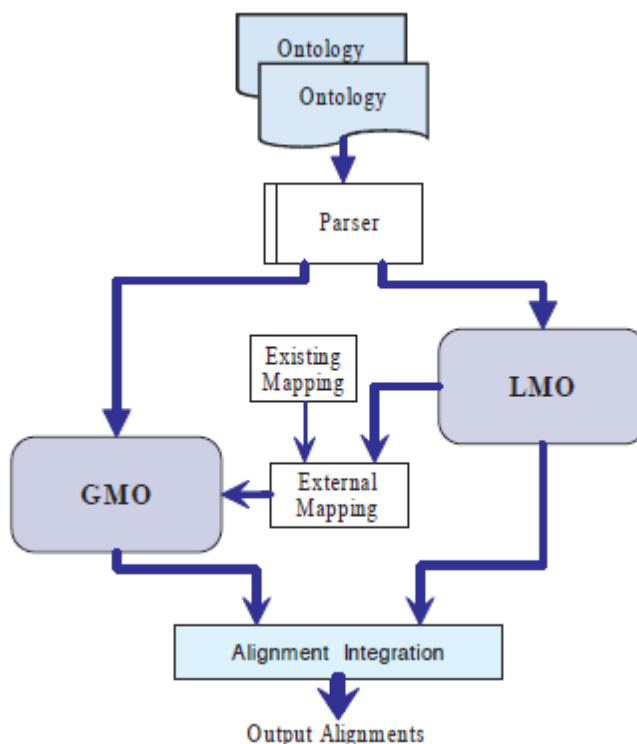


Figura 23. Arquitectura del sistema Falcon-AO (tomado de *(Jian et al., 2005)*)

Acorde a los autores, Falcon-AO obtiene buenos resultados cuando ambas ontologías comparten muy pocos aspectos de similitud léxica pero son estructuralmente muy compatibles.

Falcon-AO *(Hu & Qu, 2008)* integra cuatro tipos de algoritmos (V-Doc, I-Sub, GMO y PBM) en vez de los dos iniciales. Los resultados son representados en RDF/XML e incorpora un repositorio de alineamientos con el objetivo de minimizar recursos en sucesivas comparaciones. Los algoritmos V-Doc y I-Sub son algoritmos basados en medidas léxicas;

GMO, en similitudes estructurales y PBM divide en regiones por parecido estructural las ontologías, donde el resto de algoritmos trabajan de manera independiente.

6.1.6.SAMBO

SAMBO (*Lambrix & Tan, 2011*) es un sistema para mapear y combinar ontologías biomédicas. El uso de ontologías biomédicas ha crecido en la última década, tal ha sido su importancia que se han creado organismos para su gestión y calidad como *Gene Ontology Consortium* en 1998. Este tipo de ontologías se caracteriza por una terminología específica, por un número elevado de elementos y por ser, básicamente, taxonomías. SAMBO es un sistema que combina diferentes estrategias mediante una suma ponderada. La elección de las estrategias y los pesos recae en el usuario. Las estrategias se basan en:

- Un mapeado de terminologías, son dos técnicas léxicas como son q-gram y distancia de edición. Para aumentar la probabilidad de acierto cada palabra es pre procesada por un analizador léxico, en este caso Porter, junto con los posibles sinónimos y superclases en WordNet.

- Un mapeado estructural, es un algoritmo iterativo basado en las relaciones de «is-a» (*rdf:type*) y «part-of» (*owl:subclassOf*) en las jerarquías. Este algoritmo necesita de alineamientos previos para co-tejar su viabilidad calculando las distancias en la jerarquía mediante las dos anteriores relaciones.
- Con conocimiento del dominio, los autores en este caso utilizan el Meta tesoro de Unified Medical Language Systems (UMLS) el cual está organizado mediante conceptos que comparten sinónimos y sin-dicado. La similitud de dos términos se calcula mediante la similitud en UMLS.
- Un sistema de autoaprendizaje que consiste en la clasificación del concepto en una serie de resúmenes de publicaciones médicas. Dos conceptos son similares si comparten los mismos resúmenes.

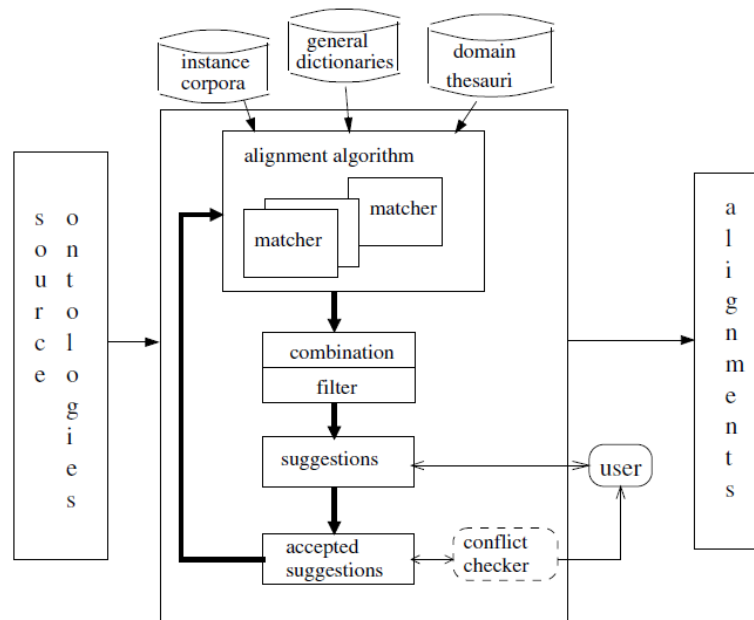


Figura 24. Estrategia de alineación del framework SAMBO
(tomado de (Lambrich & Tan, 2011))

Los autores comparan su propuesta con tres herramientas: Protégé (Noy et al., 2003), PROMPT (Noy & Musen, 2003) y FOAM (Ehrig & Sure, 2005). Los resultados de SAMBO son mejores o iguales a los del resto en las diferentes pruebas. En este estudio no está indicado el peso desempeñado por los recursos que manejan conocimiento sobre el dominio.

6.1.7.AROMA

Association Rule Ontology *Matching* Approach (AROMA) (David, 2007) es un algoritmo basado en el descubrimiento

de reglas de asociación entre diferentes elementos. Las reglas de asociación determinan la subsunción entre dos elementos y se basan en la jerarquía de la estructura. Un trabajo parecido, basado en jerarquías, por parte de los mismos autores confirma la resolución de estructuras taxonómicas (David, Guillet, Gras, & Briand, 2006). Para buscar estas correspondencias entre la estructura se utiliza a nivel léxico la medida JaroWinkler.

La idea detrás de las reglas de asociación es que un elemento estará subsumido por otros elementos si su vocabulario y descendientes están englobados por el elemento subsumidor.

AROMA está dividido en tres sucesivas etapas: (David, 2009)

1. La etapa de pre procesamiento en cada entidad como clases y propiedades, por un conjunto de términos.
2. La segunda etapa consiste en descubrir las reglas de asociación entre entidades y finalmente.
3. El post procesamiento, etapa que tiene como objetivo limpiar y mejorar el resultado de la alineación.

La primera etapa construye un conjunto de términos relevantes o valores de datos para cada clase y propiedad. Para hacer esto, se extrae el vocabulario de la clase y propiedad de sus anotaciones y valores individuales con la ayuda de un sencillo y binario extractor de términos aplicado a las cadenas de texto.

La segunda etapa de AROMA descubre las relaciones de subsunción por el uso de los modelos de reglas de asociación y la medida de implicación intensiva (*Gras, Suzuki, Guillet, & Spagnolo, 2008*).

La última etapa corresponde al post procesamiento del conjunto de reglas de asociación. Se lleva a cabo las siguientes tareas:

- Deducción de las relaciones de equivalencia.
- Supresión de los ciclos en los grafos de asociación.
- Supresión de las correspondencias redundantes.
- Selección de la mejor correspondencia para cada entidad (la alineación es una función inyectiva).
- El mejoramiento de la alineación por el uso de matcher basado en similitud y descubrimiento previo de correspondencias.

6.1.8.RIMON

RiMOM (*Li et al., 2009*) es un algoritmo que implementa la selección dinámica de estrategias según la naturaleza de las ontologías implicadas. Esta selección se basa en la variabilidad de dos tipos de medidas basadas en los nombres y en la similitud estructural las cuales ponderan los resultados de diferentes medidas básicas de similitud: léxicas y estructurales. Los autores opinan que no es siempre adecuado utilizar la misma configuración en la combinación de medidas y esta configuración ha de ser realizada de manera dinámica para flexibilizar el proceso.

Las medidas básicas son: la distancia de edición en la comparación de nombres de clases; una estrategia basada en espacio vectorial teniendo en cuenta las instancias y los comentarios de las clases; y a nivel estructural usan una variación del algoritmo de similitud *flooding* (SF) (*Melnik, Garcia-Molina, & Rahm, 2002*). En este último algoritmo la ontología se transforma en un grafo etiquetado dirigido

(directed labeled graph, DLG) según la conectividad de ambos grafos se establecen la semejanza de los nodos.

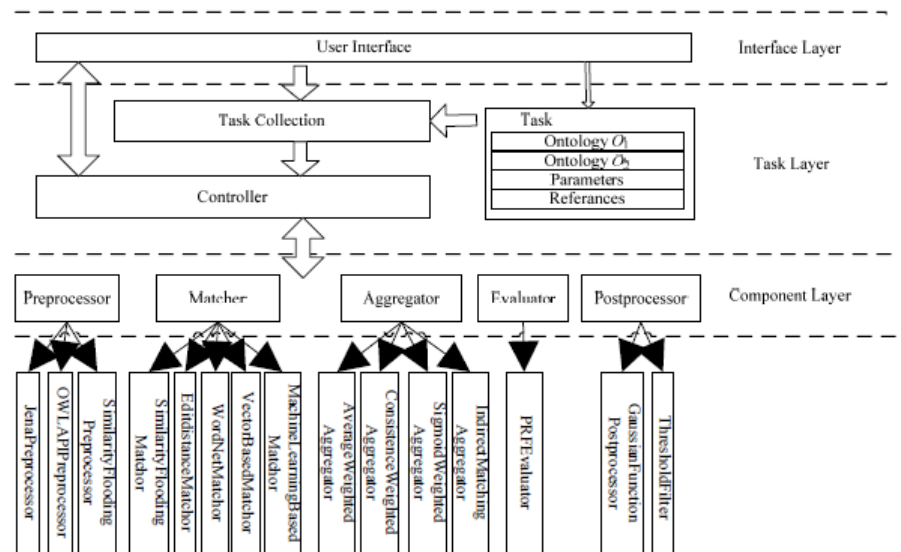


Figura 25. Arquitectura del sistema RiMOM (tomado de (Wang *et al.*, 2010))

6.1.9.AGREEMENT MAKER

AgreementMaker (Cruz et al., 2009) es un entorno con interfaz visual que da cabida a un repertorio de implementaciones de medidas básicas: léxicas y estructurales. Donde los diferentes métodos pueden combinarse o simplemente ejecutarse paralelamente. El resultado es una suma ponderada de los diferentes métodos que el usuario decida

incluir. La ponderación de cada uno de ellos puede ser manual o automática acorde a sus experimentos.

Incorporan tres nuevas medidas: similitud base, similitud de herencia por descendientes y similitud por hermandad. Las tres medidas se definieron en un trabajo anterior (*Cruz & Sunna, 2008*).

La similitud base es una función por donde todos los conceptos son tratados bajo una serie de condiciones para establecer un valor de similitud. Al superar un umbral, este valor determina un alineamiento. Los criterios son léxicos por comparación de nombres, simples o compuestos, si coinciden en nombre se determina su equivalencia inmediata, si no es así se comprueba el número de palabras coincidentes en su definición mediante el uso de WordNet. La asignación de una acepción es inexplicada.

La similitud de herencia por descendientes se basa en una heurística que tiene en cuenta la propagación en diferentes niveles de jerarquía en WordNet la similitud de las clases padres. De la misma manera, la similitud por hermandad tiene en cuenta la similitud entre las clases hermanas, comparten la clase antecedente o padre.

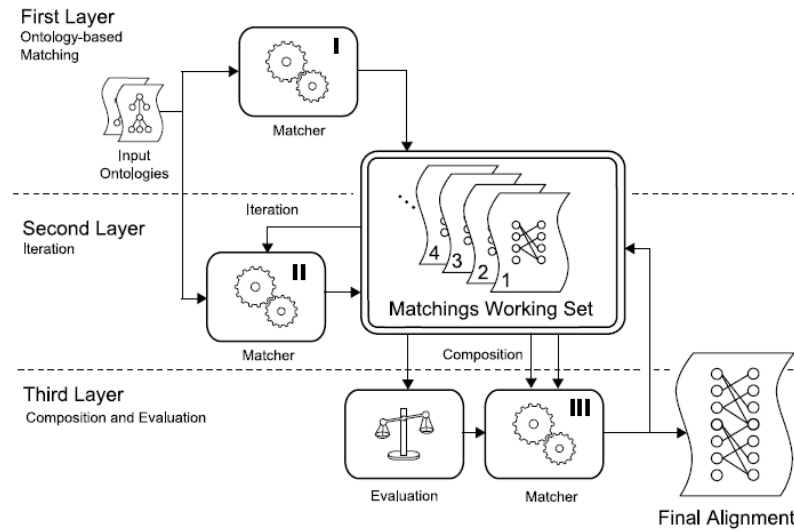


Figura 26. AgreementMaker OAEI 2010 (tomado de [\(Cruz et al., 2010\)](#))

6.1.10. ASMOV

ASMOV ([Jean-Mary, Shironoshita, & Kabuka, 2009](#)) es un algoritmo que usa similitudes léxicas y estructurales de dos ontologías para ir calculando iterativamente la similitud entre ellas y verificar que los alineamientos no contengan inconsistencias semánticas. Según los autores la naturaleza de las similitudes es similar a la propuesta realizada en OLA, donde la combinación de las diferentes medidas se basa en una suma ponderada.

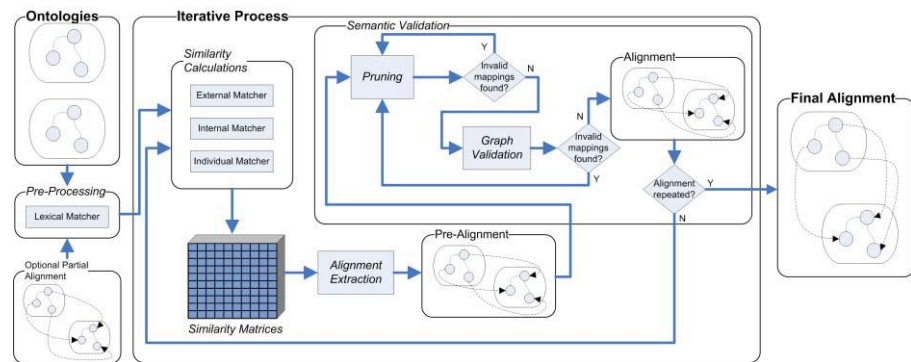


Figura 27. El proceso de mapping de ASMOV (tomado de [Jean-Mary et al., 2010](#))

Se ha definido similitud léxica, entre nombres y etiquetas, de la siguiente manera: la equivalencia, es 1, cuando dos nombres son idénticos; una similitud de 0.99 cuando un nombre es sinónimo del otro, con la peculiaridad de poder funcionar sin un tesoro externo WordNet o UMLS; una similitud de 0 cuando un nombre es antónimo del otro; y un valor de proximidad entre conceptos dentro de un tesoro acorde al trabajo de [Lin, 1998](#). En caso de palabras compuestas, la similitud es computada por el número de palabras coincidentes.

Los comentarios de las clases son procesados de manera independiente. Mediante una variación de la fórmula de Levenshtein se calcula el número de pasos necesarios para transformar una cadena a otra. Las similitudes léxicas

se agrupan mediante una suma ponderada donde los pesos son fijados experimentalmente.

A nivel estructural se computa la relación jerárquica entre clases y propiedades mediante la similitud que guardan las clases o propiedades allegadas a padres e hijos en los diferentes niveles. Una vez determinadas los alineamientos estos se someten a una serie de condiciones lógicas con el objetivo de encontrar alguna incongruencia.

6.1.11. TAXOMP

TaxoMap (*Hamdi et al., 2009*), los autores en esta segunda versión consideran una ontología como una relación taxonómica por el gran número de ontologías utilizadas para tal fin sin que haya una especificación adecuada de las clases, propiedades e instancias. Definen tres tipos de resultados: por equivalencia, por jerarquía y por «semantically related». Estas últimas, cuando se desconoce el tipo de relación. La equivalencia se establece por similitud de etiquetado. La relación de subsunción mediante la comprobación de la inclusión de la etiqueta. La comparación de nombres se realiza mediante la reducción de la etiqueta al lema, se ignoran los conectores. Además consideran la

posición de una palabra dentro de la propia etiqueta, en palabras compuestas, pero no especifican como la gestionan.

La segunda versión de Taxomap está especialmente diseñada para ser más eficiente en grandes ontologías. Los autores acorde con un algoritmo trocean la ontología y estos bloques pasan por el anterior algoritmo, consecuentemente son unidos con sus correspondientes alineamientos.

6.1.12. PROMPT

PROMPT (*Soriyan et al., 2012*) es un algoritmo desarrollado por Noy y Musen en el año 2000 para fusión y alineación ontológica. Prompt maneja ontologías específicas en formatos compatibles con OKBC. Comienza con la identificación de nombres de las clases que coincidan. Basado en esto, un método iterativo se lleva a cabo para realizar actualizaciones automáticas encontrando conflictos resultantes, y haciendo sugerencias para eliminar estos conflictos. PROMPT está implementado como una extensión de Protégé 2000 (*Noy et al., 2003*) herramienta de adquisición de conocimiento y ofrece una colección de operaciones

para fusión de dos clases y slots relacionados. Prompt no provee una instancia o estructura de *matching* y no se ocupa de procesos de normalización.

6.1.13. DSSIM

DSSIM es un framework para *matching* ontológico basado en agentes. El sistema maneja Ontologías a gran escala en OWL y SKOS y calcula alineaciones 1:1 con relación de equivalencias y subsunción entre conceptos y propiedades. El framework utiliza la teoría de Dempster-Shafer ([Shafer, 1976](#)) en el contexto de preguntas y respuestas ([Nagy & Vargas-Vera, 2010](#)), ([Nagy et al., 2008](#)), específicamente, cada agente crea un valor de confianza para la corrección de una hipótesis de correspondencia. Entonces, estos valores de confianza son combinados en un solo y más coherentes vista para mejorar la calidad de la correspondencia.

La ontología es inicialmente particionada en fragmentos, cada concepto o propiedad de una primera ontología particionada es vista como una consulta, la cual, es expandida basada en hipónimos de WordNet, ([Miller, 1995](#)) visto como el conocimiento previo. Estos hipónimos son usados

como variables en la hipótesis para mejorar los valores de confianza. Los conceptos y propiedades expandidos son emparejados sintácticamente con el concepto o propiedad similar en la segunda ontología con el propósito de identificar un fragmento de grafo correspondiente en la segunda ontología. Entonces, el grafo de consulta de la primera ontología es emparejado con el fragmento de grafo relevante de la segunda ontología, para esto, varias medidas terminológicas son usadas, como la distancia Jaccard y Monger-Elkan, que son combinadas usando las reglas de Dempster.

Cuando se presentan conflictos de valores de creencia son resueltos usando un enfoque de votación *Fuzzy* equipada con cuatro reglas ad-hoc¹⁰ (si-entonces).

El sistema no tiene una interfaz gráfica, pero usa el sistema AQUA (Lopez, Pasin, & Motta, 2005) de preguntas y respuestas, habilitado para manejo de preguntas en lenguaje natural.

¹⁰ Una hipótesis ad hoc es una hipótesis concreta creada para explicar un hecho que contradice una teoría

6.2. DESCRIPCIÓN DE TÉCNICAS DE *MATCHING* BÁSICAS

6.2.1. TÉCNICAS SINTÁCTICAS BASADAS EN CADENAS

En esta etapa se busca determinar una similitud entre los elementos, estos elementos son las entidades ontológicas, para esto necesitamos considerar la mínima unidad de representación, que son las palabras, que definen los conceptos, unidades léxicas. En el nivel léxico podemos encontrar el tipo de medida que compara la similitud entre caracteres.

La similitud es la función de un par de entidades a un número real que expresa la similitud, la cual para el cálculo se utiliza un valor normalizado entre 0 y 1, donde 0 significa que las dos entidades no tienen nada en común y 1 si las cadenas son idénticas.

Para realizar la comparación entre cadenas primero se debe realizar un pre-procesamiento a las entidades de entrada, este pre-procesamiento puede ser un proceso de normalización, en el cual se utilizan diversas técnicas como:

- Normalización de minúsculas: consiste en convertir cada letra en su igual, pero en minúscula.

PIZZA → pizza

- Supresión de tildes: es la eliminación de cualquier virgulilla o rasgo que se coloca sobre alguna letra, en español se refiere al acento ortográfico.

Canción → cancion

- Normalización de espacios: este consiste en eliminar todos los espacios en blanco como tabulaciones, retornos de carro, espacios en blanco y reducir a solo uno.

Ciencia·politica → ciencia·politica

- Supresión de guiones: consiste en eliminar algunos vínculos o enlaces entre palabras y sustituirlos por espacios en blanco.

Pre-procesamiento → pre procesamiento

- Supresión de número: consiste en eliminar números contenidos entre las palabras.

Cod5895 → cod

- Supresión de la puntuación: consiste en eliminar signos como dos punto, punto y coma, coma, punto, etc.

Ing. Sistemas → ing sistemas

Para aplicar el proceso de normalización se debe tener cuidado especial en no eliminar semántica al momento de aplicarlo, eliminar tildes puede hacer perder significado, por ejemplo, te y té. También al eliminar números de las palabras, por ejemplo, carbono-14.

6.2.2.DISTANCIA ENTRE CADENAS

Medida que nos dice cuántas letras se deben cambiar para convertir una cadena en otra, la función más utilizada es la distancia de Hamming (*Hamming, 1950*). Permite determinar la diferencia entre dos palabras mediante el cambio entre caracteres de una palabra para obtener la otra.

$$\delta(s, t) = \frac{\left(\sum_{i=1}^{\min(|s|, |t|)} s[i] \neq t[i] \right) + ||s| - |t||}{\max(|s|, |t|)}$$

Otra función muy utilizada es la distancia de Levenshtein (también llamada distancia de edición). En la distancia de edición se consideran operación de inserción, reemplazo y borrado de un carácter, a cada operación se le asigna un costo y la distancia entre dos cadenas es la suma del costo de cada operación aplicada para obtener la segunda

cadena. La distancia de Levenshtein (*Levenshtein, 1966*) es el mínimo número de inserciones, borrado y reemplazo de caracteres necesarios para transformar una cadena en otra.

$$\delta(s, t) = \min \left(\sum op_i(s) = t \right)$$

Distancia de Jaro (*Jaro, 2007*). Dada dos palabras $s = a_1, \dots, a_k$ y $t = b_1, \dots, b_L$ se define un carácter común si $b_j = a_i$ tal que $i - H \leq j \leq i + H$ donde $H = \frac{\min(|s|, |t|)}{2}$. Siendo $s' = a'_1 \dots a'_k$ el conjunto de caracteres en común en t (en el mismo orden que aparecen en s) y siendo análogo $t' = b'_1 \dots b'_L$, se define una transposición por s' , t' en una posición i tal que $a'_i \neq b'_i$. Siendo $T_{s', t'}$ la mitad del número de transposiciones para s' y t' . La similitud de Jaro para s y t es:

$$\delta(s, t) = \frac{1}{3} \left(\frac{|s'|}{|s|} + \frac{|t'|}{|t|} + \frac{|s'| - T_{s', t'}}{|s'|} \right)$$

La distancia de Jaro-Winkler (*Winkler, 1999*) es una variante de la distancia de Jaro, donde se usa una longitud P del prefijo común más largo entre s y t , siendo $P = \max(P, 4)$ se define:

$$\delta(s, t) = \delta_{jaro}(s, t) + \frac{P'}{10} \left(1 - \delta_{jaro}(s, t) \right)$$

Similitud de Jacard (*Jaccard, 1908*) se define como el número de elementos de la intersección dividido por el total de elementos de la unión. Esta similitud se aplica en este caso sobre caracteres.

$$\delta(s, t) = \frac{|s \cap t|}{|s \cup t|}$$

Coeficiente de Dice (*Dice, 1945*) se define como la información compartida entre dos elementos sobre la cardinalidad de ambos

$$\delta(s, t) = \frac{2|s \cap t|}{|s| + |t|}$$

6.2.3.SIMILITUD ENTRE SUBCADENAS

Estas pruebas consisten en comparar subcadenas de un par de cadenas y así determinar un grado de correlación, diferentes variaciones pueden ser obtenidas a partir de la igualdad de cadenas.

La prueba de subcadena es una similitud que mide la proporción de la parte común entre dos cadenas.

$$\delta(s, t) = \frac{2|x|}{|s| + |t|}$$

Donde s y t son cadenas y x es la cadena común más larga de s y t ;

Una similitud de prefijo o sufijo puede ser definida de este modelo a partir de pruebas de sufijo y prefijo, que probarían cuando una cadena es prefija o sufija de otra.

La similitud n-gram (Cavnar & Trenkle, 1994), otra de las pruebas la cual calcula el número de n-grams comunes, es decir, secuencias de n caracteres entre ellas, por ejemplo, los tri-grams de la cadena *ontología* son:

$$\begin{aligned} ngram(ontologia, 3) &= \{ont, nto, tol, olo, log, ogi, gia\} \\ \delta(s, t) &= |ngram(s, n) \cap ngram(t, n)| \end{aligned}$$

La cual se puede normalizar así:

$$\delta(s, t) = \frac{|ngram(s, n) \cap ngram(t, n)|}{\min(|s|, |t|) - n + 1}$$

Donde $ngram(s, n)$ es el conjunto de ngramas obtenido de la cadena s de longitud n .

6.2.4.TÉCNICAS SINTÁCTICAS BASADOS EN RES- TRICCIONES

Estos métodos están basados en la estructura interna de las entidades y usan criterios como el conjunto de sus propiedades, el rango de sus propiedades (atributos y relaciones), su cardinalidad, la transitividad, etc., para calcular la similitud entre ellos.

Estos métodos son usados para crear agrupamientos de correspondencias ya que existen numerosas entidades que comparten dominios y rangos similares, también son combinados con otras técnicas a nivel de elemento para mejorar su precisión.

6.2.5.COMPARACIÓN DE TIPOS DE DATOS

Se usan reglas para realizar el casting de una entidad de un tipo para convertir la entidad en otro tipo, estas reglas tienen cierto grado de correspondencia dependiendo la compatibilidad entre tipos.

	String	Booleano	Decimal	Float	Double	Datetime	Time	date	Char
String	1.0	0.2	0.4	0.3	0.3	0.1	0.1	0.1	0.7
Booleano		1.0	0.1	0.0	0.0	0.0	0.0	0.0	0.2
Decimal			1.0	0.6	0.6	0.0	0.0	0.0	0.2
Float				1.0	0.9	0.0	0.0	0.0	0.1
Double					1.0	0.0	0.0	0.0	0.1
Datetime						1.0	0.5	0.5	0.0
Time							1.0	0.5	0.0

date								1.0	0.0
Char									1.0

Tabla 13. Tabla de compatibilidad de tipos

6.2.6.COMPARACIÓN DE DOMINIO

El dominio es una propiedad en clases y valores en individuos el cual puede ser estructurado en conjuntos o en secuencias, este conjunto de valores es llamado cardinalidad o multiplicidad. Esta técnica compara los dominios de las entidades a analizar teniendo diferentes criterios de valoración, entre los más usados está el uso de tablas de valores o el cálculo propuesto por (Valtchev & Euzenat) el cual los tipos o dominios de propiedades deben ser comparados sobre la base de su interpretación: conjunto de valores, La distancia entre dos dominios está dada por la diferencia entre su tamaño y el de su generalización común. Esta medida suele ser normalizada por el tamaño de la mayor distancia posible con un tipo de dato particular.

$$\delta(d, d') = \frac{|gen(d \cup d')| - |gen(d \cap d')|}{|TD|}$$

Donde d y d' son dominios sobre un tipo de dato TD, y gen(x) es la generalización de un tipo de expresión.

6.2.7.TECNICAS SINTACTICAS A NIVEL DE ESTRUCTURA

Una ontología puede ser considerada como un grafo cuyas aristas están etiquetadas por los nombres de las relaciones, considerando los ciclos en las relaciones de subsunción.

En este trabajo se tendrá en cuenta una medida estructural, la que no emplea recursos externos, basándose en técnicas de comparación de árboles y mayoritariamente de grafos. La idea con estas medidas es determinar el grado de similitud entre estructuras cercanas a los conceptos, haciendo uso de procesos recursivos.

Recordemos que el modelo conceptual utilizado en esta tesis es el propuesto por Chavarro en ([Chavarro, 2012](#)), el cual utiliza un Grafárbol ontológico para representar una ontología. Vea [sección 3.5.2](#), y en la cual se modelan dos tipos de relaciones entre entidades e individuos, una estructura jerárquica formada por la relación «es un», una estructura de composición formada por las relaciones funcionales «es parte de» y una estructura funcional formada por las relaciones funcionales de asociación.

SIMILITUD DE TAXONOMÍA

(Rada, Mili, Bicknell, & Blettner, 1989) Una medida genérica posible para determinar la similitud semántica de conceptos c dentro de una ontología, en la jerarquía de conceptos, es la similitud de taxonomía, que es una función tal que:

$$\delta(c, c') = \begin{cases} e^{-\alpha l \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}}, & \text{si } c \neq c' \\ 1, & \text{en otro caso} \end{cases}$$

Donde α, β son parámetros escalares y deben ser mayores a cero, l es la longitud del camino más corto entre c, c' y h es la profundidad en la jerarquía de conceptos.

7

7. BIBLIOGRAFÍA

Aumüller, D., Do, H. H., Massmann, S., & Rahm, E. (2005, 2005). *Schema and Ontology Matching with COMA++*.

Batini, C., Lenzerini, M., & Navathe, S. B. (1986). A Comparative Analysis of Methodologies for Database Schema Integration. *ACM computing surveys (CSUR)*, 18(4), 323-364.

Berners-Lee, T. (2000). *Weaving the Web*. New York: Harper Business.

Biron, P., Malhotra, A., & Consortium, W. W. W. (2004). XML Schema Part 2: Datatypes. *World Wide Web*

Consortium Recommendation REC-xmlschema-2-20041028.

Cavnar, W. B., & Trenkle, J. M. (1994). N-gram-based text categorization. *Ann Arbor MI*, 48113(2), 161-175.

CERN. (2017). About CERN. Retrieved from <https://home.cern/about>

Chavarro, J. C. (2012). *Marco de Referencia para la Gestión del Cambio en Ontologías para la Web Semántica: Búsqueda, Reutilización, Mezcla e Integración*. Tesis Doctoral. Universidad del Valle.

Chavarro, J. C., & Millán, M. E. (2010). *Modelado Conceptual para la Gestion del Cambio Ontológico*. Paper presented at the Quinto Congreso Colombiano de Computación, Cartagena.

Choi, N., Song, I. Y., & Han, H. (2006). A Survey on Ontology Mapping. *ACM Sigmod Record*, 35(3), 34-41.

Cruz, I. F., Antonelli, F. P., & Stroe, C. (2009). AgreementMaker: Efficient Matching for Large Real-world Schemas and Ontologies. *Proceedings of the VLDB Endowment*, 2(2), 1586-1589.

Cruz, I. F., Stroe, C., Caci, M., Caimi, F., Palmonari, M., Antonelli, F. P., & Keles, U. C. (2010, 2010). *Using AgreementMaker to Align Ontologies for OAEI 2010*.

Cruz, I. F., & Sunna, W. (2008). Structural Alignment Methods with Applications to Geospatial Ontologies. *Transactions in GIS*, 12(6), 683-711.

David, J. (2007). Association Rule Ontology Matching Approach. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(2), 27-49.

David, J. (2009, 2009). *AROMA Results for OAEI 2009*.

David, J., Guillet, F., Gras, R., & Briand, H. (2006). Conceptual Hierarchies Matching: An Approach Based on Discovery of Implication Rules Between Concepts. *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS*, 141.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3), 297-302.

- Do, H. H., & Rahm, E. (2002, 2002). *COMA: A System for Flexible Combination of Schema Matching Approaches*.
- Doan, A. H., Madhavan, J., Domingos, P., & Halevy, A. (2004). Ontology Matching: A Machine Learning Approach. *Handbook on Ontologies in Information Systems*, 397-416.
- Ehrig, M. (2007). *Ontology Alignment: Bridging the Semantic Gap*. : Springer.
- Ehrig, M., & Sure, Y. (2005, 2005). *FOAM-framework for Ontology Alignment and Mapping-Results of the Ontology Alignment Evaluation Initiative*.
- Euzenat, J. (2001). Towards a Principled Approach to Semantic Interoperability. *Proc. IJCAIWorkshop on Ontologies and Information Sharing*, 19–25.
- Euzenat, J., Ferrara, A., Hollink, L., Isaac, A., Joslyn, C., Malaisé, V., . . . Sabou, M. (2010). Results of the Ontology Alignment Evaluation Initiative 2009.

Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., & Trojahn, C. (2011). Ontology Alignment Evaluation Initiative: Six Years of Experience. *Journal on data semantics XV*, 158-192.

Euzenat, J., & Shvaiko, P. (2007). *Ontology Matching* (Vol. 18): Springer Berlin.

Fernández, A. (2004). *Extracción de Información de la Web Basado en Ontologías*. (Maestria en Ingenieria en Computacion Tesis de Maestria), Universidad de la Republica, Montevideo, Uruguay.

Flouris, G., Plexousakis, D., & Antoniou, G. (2006, 2006). *A Classification of Ontology Change*.

Gangemi, A., Pisanelli, D., & Steve, G. (1998, 1998). *Ontology integration: Experiences with Medical Terminologies*.

Giunchiglia, F., Shvaiko, P., & Yatskevich, M. (2004). S-Match: An Algorithm and an Implementation of Semantic Matching. *The semantic web: research and applications*, 61-75.

Gómez, H. (2008). *Cobber: Un Enfoque Sistémico, Afectivo y Ontológico para el Razonamiento Basado en Casos Conversacional*. Universidad Complutense de Madrid. Retrieved from <http://dialnet.unirioja.es/servlet/tesis?codigo=20038> Available from Google Scholar

Granitzer, M., Sabol, V., Onn, K. W., Lukose, D., & Tochtermann, K. (2010). Ontology Alignment—A Survey with Focus on Visually Supported Semi-automatic Techniques. *Future Internet*, 2(3), 238-258.

Gras, R., Suzuki, E., Guillet, F., & Spagnolo, F. (2008). *Statistical Implicative Analysis: Theory and Applications* (Vol. 127): Springer.

Gruber, T. R. (1995). Toward Principles for the Design of Ontologies used for Knowledge Sharing. *International journal of human computer studies*, 43(5), 907-928.

Guarino, N. (1998). Formal Ontology and Information Systems. *Proceeding of FOIS '98*, Trento.

Hamdi, F., Safar, B., Niraula, N., & Reynaud, C. (2009, 2009). *TaxoMap in the OAEI 2009 Alignment Contest*.

- Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell System technical journal*, 29(2), 147-160.
- Howe, D. (1993). Free On-Line Dictionary of Computing. Retrieved from <http://foldoc.org/>
- Hu, W., & Qu, Y. (2008). Falcon-AO: A Practical Ontology Matching System. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3), 237-239.
- Jaccard, P. (1908). *Nouvelles recherches sur la distribution florale*.
- Jaro, M. A. (2007). Probabilistic linkage of large public health data files. *Statistics in medicine*, 14(5-7), 491-498.
- Jean-Mary, Y. R., Shironoshita, E. P., & Kabuka, M. R. (2009). Ontology Matching with Semantic Verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3), 235-251.

Jean-Mary, Y. R., Shironoshita, E. P., & Kabuka, M. R. (2010). Asmov: Results for OAEI 2010. *Ontology Matching*, 126.

Jian, N., Hu, W., Cheng, G., & Qu, Y. (2005, 2005). *Falcon-ao: Aligning Ontologies with Falcon*.

Kalfoglou, Y., & Schorlemmer, M. (2003). Ontology Mapping: The State of the Art. *The knowledge engineering review*, 18(1), 1-31.

Kashyap, V., & Sheth, A. (1997). Semantic Heterogeneity in Global Information Systems. *Cooperative Information Systems: Current Trends & Directions*", Eds: M. Papazoglou and G. Schlageter.

Kengue, J. F. D., Euzenat, J., & Valtchev, P. (2007, 2007). *OLA In the OAEI 2007 Evaluation Contest*.

Klein, M. (2004). *Change Management for Distributed Ontologies*: sn.

Klein, M., & Fensel, D. (2001, 2001). *Ontology Versioning on the Semantic Web*.

- Klein, M., Fensel, D., Kiryakov, A., & Ognyanov, D. (2002). Ontology Versioning and Change Detection on the Web. *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, 247-259.
- Lambrix, P., & Tan, H. (2011). SAMBO: A System for Aligning and Merging Biomedical Ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(3).
- Larson, J. A., Navathe, S. B., & Elmasri, R. (1989). A Theory of Attributed Equivalence in Databases with Application to Schema Integration. *Software Engineering, IEEE Transactions on*, 15(4), 449-463.
- Lassila, O., & McGuinness, D. (2001). The Role of Frame-Based Representation on the Semantic Web. *Linköping Electronic Articles in Computer and Information Science*, 6(5).
- Lassila, O., & Swick, R. R. (1998). Resource Description Framework (RDF) Model and Syntax Specification.

Levenshtein, V. I. (1966, 1966). *BINARY CODES CAPABLE OF CORRECTING DELETIONS AND INSERTIONS AND REVERSALS.*

Li, J., Tang, J., Li, Y., & Luo, Q. (2009). RiMOM: A Dynamic Multistrategy Ontology Alignment Framework. *Knowledge and Data Engineering, IEEE Transactions on*, 21(8), 1218-1232.

Lin, D. (1998, 1998). *An information-Theoretic Definition of Similarity.*

Lopez, V., Pasin, M., & Motta, E. (2005). Aqualog: An ontology-Portable Question Answering System for the Semantic Web. *The semantic web: research and applications*, 135-166.

Mahesh, K. (1996). *Ontology Development for Machine Translation: Ideology and Methodology*: Citeseer.

Masinter, L., Berners-Lee, T., & Fielding, R. T. (2005). *Uniform Resource Identifier (URI): Generic Syntax.*

McGuinness, D. L., & Van Harmelen, F. (2004). OWL Web Ontology Language Overview. *W3C recommendation*, 10(2004-03).

Melnik, S., Garcia-Molina, H., & Rahm, E. (2002, 2002). *Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching*.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), 39-41.

Nagy, M., & Vargas-Vera, M. (2010). Towards an Automatic Semantic Data Integration: Multi-Agent Framework Approach. *Chapter in Semantic Web. In-Tech Education and Publishing KG*.

Nagy, M., Vargas-Vera, M., Stolarski, P., & Motta, E. (2008, 2008). *DSSim Results for OAEI 2008*.

Noy, N. F., Crubézy, M., Fergerson, R. W., Knublauch, H., Tu, S. W., Vendetti, J., & Musen, M. A. (2003, 2003). *Protege-2000: An Open Source Ontology Development and Knowledge Acquisition Environment*.

Noy, N. F., & Klein, M. (2002). Ontology Evolution : Not the Same as Schema Evolution. *Tech Report. SMI-2002-0926, Stanford Medical Informatics.*

Noy, N. F., & Musen, M. A. (2003). The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping. *International Journal of Human-Computer Studies*, 59(6), 983-1024.

O'Reilly, T. (2005). Design Patterns and Business Models for the Next Generation of Software. *conference brainstorming session between O'Reilly and MediaLive International.*

OMG. (2007). Ontology Definition Metamodel.
<http://www.omg.org/docs/ptc/07-09-09.pdf>

Parent, C., & Spaccapietra, S. (1998). Issues and Approaches of Database Integration. *Communications of the ACM*, 41(5es), 166-178.

Pérez, M. C. (2005). Estudios de Lingüística Española. *Universidad de Málaga.*

Pingdom. (2016). Internet 2015 in Numbers. Retrieved from <http://royal.pingdom.com/2013/01/16/internet-2015-in-numbers/>

Powell, S. (2007). Cuestión Semántica. *HSM Management*, 1(5), 86.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1), 17-30.

Rahm, E., & Bernstein, P. A. (2001). A Survey of Approaches to Automatic Schema Matching. *the VLDB Journal*, 10(4), 334-350.

Salazar, D. (2015). *Linea Base Arquitectural del Framework OntoConcept*. (Proyecto de Grado).

Sánchez, I. N., & Béjar, R. M. (2000). *Teoría de Autómatas y Lenguajes Formales*: Diego Marín.

Senso, J. A., & De la Rosa Piñero, A. (2003). El concepto de metadato. Algo más que descripción de recursos electrónicos. *Ci. Inf.*, 32, 95-106.

Senso, J. A., & Piñero, A. R. (2003). El Concepto de Metadato. Algo Más que Descripción de Recursos Electrónicos. *Ciência da informação*, 32(2), 95–106
<http://www.scielo.br/pdf/ci/v132n102/17038.pdf/>

Shafer, G. (1976). *A Mathematical Theory of Evidence* (Vol. 76): Princeton university press Princeton.

Shvaiko, P., & Euzenat, J. (2005). A Survey of Schema-Based Matching Approaches. *Journal on Data Semantics IV*, 146-171.

Shvaiko, P., Giunchiglia, F., & Yatskevich, M. (2010). Semantic Matching with S-Match. *Semantic Web Information Management: a Model-Based Perspective*, 20, 183-202.

Smith, M. K., Welty, C., & McGuinness, D. L. (2004). *OWL Web Ontology Language: Guide*. Retrieved from

Soriyan, A., Gambo, I., & Ikono, R. (2012). State-of-the Art: A Comparative Analysis of Ontology Matching

Systems. *African Journal of Computing & ICT*, 5(4).

Staab, S., & Studer, R. (2009). *Handbook on Ontologies*: Springer.

Stats, I. L. (2016). Internet Live Stats. Retrieved from <http://www.internetlivestats.com/>

Stojanovic, L., Maedche, A., Motik, B., & Stojanovic, N. (2002). *User-Driven Ontology Evolution Management*. Paper presented at the Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web. <http://www.springerlink.com/index/CGU1LTK11Y3NJTYN.pdf>

Strawson, P. F., & Bubner, R. (1975). *Semantik und Ontologie*: Vandenhoeck & Ruprecht.

Studer, R., Benjamins, R., & Fensel, D. (1998). Knowledge Engineering Principles and Methods. *Data and Knowledge Engineering*, 161-197.

Valtchev, P., & Euzenat, J. (2013/02/23/22:18:04). Construction automatique de taxonomies pour l'aide à la représentation de connaissances par objets = Automatic taxonomy design for object-based knowledge representations. Retrieved from

<http://cat.inist.fr/?aModele=afficheN&cpsidt=208488>

Wache, H., Voegelé, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., & Hübner, S. (2001, 2001). *Ontology-Based Integration of Information a Survey of Existing Approaches*.

Wang, Z., Zhang, X., Hou, L., Zhao, Y., Li, J., Qi, Y., & Tang, J. (2010, 2010). *RiMOM Results for OAEI 2010*.

Wikipedia. (2009). Metamodelado. Retrieved from <https://es.wikipedia.org/wiki/Metamodelado>

Winkler, W. E. (1999). The state of record linkage and current research problems.